



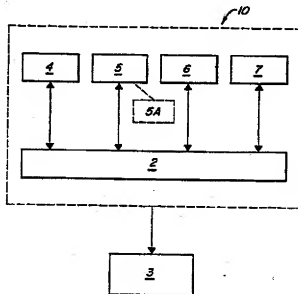
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵: G06F 3/03, 9/00, 15/18	A1	(11) International Publication Number: WO 91/17494 (43) International Publication Date: 14 November 1991 (14.11.91)																								
(21) International Application Number: PCT/US91/02890 (22) International Filing Date: 26 April 1991 (26.04.91) (30) Priority data: <table border="0"> <tr> <td>516,822</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,247</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,250</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,128</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,815</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,814</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,248</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> <tr> <td>516,813</td> <td>27 April 1990 (27.04.90)</td> <td>US</td> </tr> </table>		516,822	27 April 1990 (27.04.90)	US	516,247	27 April 1990 (27.04.90)	US	516,250	27 April 1990 (27.04.90)	US	516,128	27 April 1990 (27.04.90)	US	516,815	27 April 1990 (27.04.90)	US	516,814	27 April 1990 (27.04.90)	US	516,248	27 April 1990 (27.04.90)	US	516,813	27 April 1990 (27.04.90)	US	(72) Inventors: BACHMAN, Charles, W.; 11 Percy Road, Lexington, MA 02173 (US). ABRAMOVICH, Igor; 76 Ship Avenue #10, Medford, MA 02155 (US). GANE, Christopher, P.; 333 E. 30th Street, Apt. #10J, New York, NY 10016 (US). KRIEGER, David, A.; 40 Granville Road, Cambridge, MA 02138 (US). MICCO, John, T.; 95 Montclair Ave., Waltham, MA 02154 (US). CIMBAL, John, J.; 24 Eaton Road West, Framingham, MA 01701 (US). (74) Agents: LAPPIN, Mark, G. et al.; Lahive & Cockfield, 60 State Street, Boston, MA 02109 (US). (81) Designated States: AT (European patent), BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).
516,822	27 April 1990 (27.04.90)	US																								
516,247	27 April 1990 (27.04.90)	US																								
516,250	27 April 1990 (27.04.90)	US																								
516,128	27 April 1990 (27.04.90)	US																								
516,815	27 April 1990 (27.04.90)	US																								
516,814	27 April 1990 (27.04.90)	US																								
516,248	27 April 1990 (27.04.90)	US																								
516,813	27 April 1990 (27.04.90)	US																								
(71) Applicant: BACHMAN INFORMATION SYSTEMS, INC. [US/US]; 8 New England Executive Park, Burlington, MA 01803 (US).		Published <i>With international search report.</i>																								

(54) Title: DYNAMIC INFORMATION MANAGEMENT COMPUTER SYSTEM**(57) Abstract**

A computer system (10) for dynamically modeling information systems of organizations. The system (10) includes a plurality of editors (4-7) which enable a user to interact with the system, and an adaptive window system (3). Included in the computer system is an information modeler (4), a process modeler (5), a function modeler (7), and a logic modeler (6) for creating, analyzing and modifying design data associated with each modeler, and a filter for creating a subset of a model established by a modeler. Each of the modelers are dynamically interlinked, such that a

change to one design data set (40) will affect a corresponding change in other design data sets (40). The system includes graphic representations (50) associated with each of the modelers to enable a user to interact with the system. The objects (60, 62) of each of the graphic representations may also be dynamically linked, such that a change in one representation affects a substantially immediate change in other representations. The system further enables a user to "explode" a process (56) from a graphical representation, or diagram, into its corresponding subprocesses (58, 60, 62).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brasil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TO	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark				

DYNAMIC INFORMATION MANAGEMENT COMPUTER SYSTEM

5

BACKGROUND

This invention is directed to a computer system for dynamically modeling information systems of organizations and building models of complex
10 business transactions and creating and manipulating subprocesses and subsets of dynamic information system models of organizations.

Early computer information processing
15 systems stored, retrieved, and modified data contained in files specifically created for each information processing application. Knowledge relating to the structure and content of each data file was coded directly into each application
20 program. When a need later developed to use the data within the data file for additional purposes, the user often created other primarily redundant data files, writing a specific application program to interact with each of these files. Subsequent
25 changes in the content of any data file frequently mandated modification of related data files. If the user wished to avoid creating redundant data files, those modifications necessitated by additional demands were made to the original data file. Again,
30 application programs were written to perform new tasks on the modified file.

However, the data file modifications sometimes hampered operation of earlier application

programs, thus requiring their modification. As a consequence, computer installations involved in data processing faced the recurrent task of choosing between data file redundancy and repeated software
5 modifications.

With the advent of so-called data base management systems, many of the problems encountered in the early information processing systems were
10 alleviated. In addition to removing the need for the application program to address the intricacies of data file manipulation, data base management systems promoted efficient design, enhanced file maintenance and modification, eliminated data file redundancy,
15 and provided substantial documentation regarding data file structure.

With the shift to data base management systems, the concept of dual data representation
20 fully emerged. The first level of data representation, the physical representation, relates to the manner in which individual data records are stored and how their inter-relationships are depicted. The second level of data representation,
25 the logical representation, describes the data base users' view of the data. The physical representation of the data is generally of no concern to the data base user. Rather, the task of manipulating individual data storage areas is left to the data
30 base management system.

Data variables are named storage structures which may be used to store information, and which are basic features of information systems. These

variables are capable of being assigned a value, holding that value as long as the data variable survives, and returning that value one or more times upon request. The current value of a data variable

5 essentially remains unchanged until some explicit action is taken to change the content of the variable. These data variables have a number of applications, such as being the attributes of entities, being independent global attributes, being

10 variables of processes, and being variables of operating systems, the values of which are available to all processes. While data variables are capable of storing various types of information, each is generally specialized to handle one type of data,

15 such as character strings, integers, or real numbers.

At the first level, data variables may be used to store information. Data variables are named storage structures, and are a basic feature of

20 information systems. These variables are capable of being assigned a value, holding that value as long as the data variable survives, and returning that value once or more times upon request. The current value of a data variable essentially remains unchanged

25 until some explicit action is taken to change the content of the variable. These data variables have a number of applications, such as being the attributes of entities, being independent global attributes, being variables of processes, and being variables of

30 operating systems, the values of which are available to all processes. While data variables are capable of storing various types of information, each is generally specialized to handle one type of data, such as character strings, integers, or real numbers.

Of concern to the data base user, however, is the logical representation of the data, since the users' ability to store, retrieve, and modify aggregations of data items, data records, and data relationships is dependent upon the form in which the data base management system presents data to the user.

Information management systems handle complex environments often consisting of hundreds of elements and relationships, permitting users to manipulate and employ data in ways not always anticipated by systems designers. Representation of such elements and relationships to a user presents a unique set of problems not encountered, and certainly not resolved by present data base management systems. Instead of being organized into application-oriented files, which are always addressed in the same way, as in data base systems, the information is organized so that it can be addressed in a variety of different ways, and can be used to answer a diversity of queries. All the interesting facts about entities are stored together rather than merely those facts needed for one application. As pointed out by J. Martiñ, "Computer Data-Base Organization", 2d ed., chapter 3, pp. 24-25 (1977), this "reservoir" concept of information systems is much easier to conceive than to implement. It is a complex and lengthy operation to build up such data bases, and it is expensive to search sufficiently quickly to give real-time answers to unanticipated queries.

The use of diagrams and windows is one way to depict the intricate relationships among entities,

attributes, and other data elements in a complex information model. A variety of "windows" have been used in application programs to facilitate user interaction with a computer system. Windows have
5 been used to show a user a "menu" of selection of operations which may be performed. Windows have also been used for displaying error messages. However, windows are not anticipatory, i.e. are not able to provide a tailored selection of options available to
10 a user at a particular point in time or text.

The management of information has evolved into the field of Computer Aided Systems Engineering (CASE). As described by C. Gane, "Computer-Aided
15 Software Engineering: The Methodologies, the Products, the Future" (1988), some of the common characteristics among CASE products are that they internally build a design database "at a higher level than code statements or physical data element
20 definitions," and that they "typically hold[s] information about the data to be stored in the system, the business logic of the processes to be implemented, their physical layout of screens and reports, and other requirements/design information."
25

There are several such CASE products presently available. Two such products are EXCELERATOR, from Index Technology, Cambridge, Massachusetts, and IEW, from Knowledgeware, Inc.,
30 Atlanta, Georgia. EXCELERATOR enables an operator to work with data flow diagrams and data model diagrams, structure diagrams. An operator may also modify, delete, copy, rename, list or add a diagram. Diagram

objects may be user defined, and are stored in a dictionary.

In a similar manner, IEW enables a user to
5 work with, and create or change objects on diagrams,
which affects an encyclopedia but not existing
displays or screens. In addition, IEW can model
relationships, entities and attributes together, or
entities alone, but does not model attributes alone.
10 It is often useful to manipulate attributes
independent of entities to which they may later be
tied.

Since one of the attractions of CASE for
15 managing information systems is the availability of
graphical tools for analysis, it would be important
for a user to be able to graphically work in
"real-time", i.e. the ability to make changes to
diagram objects, and see a corresponding change to
20 corresponding diagrams on existing screens. Such
dynamic linking of objects and diagrams would then
become an important aspect of an effective
information management system utilizing CASE
technology.

25

In conjunction with graphical tools for
analysis, a user may desire to work with only a
portion or subset of an entire complex model. A user
may want to test objects on a subset of a model
30 before integrating such objects into the entire
model, while still retaining the ability to work with
the subset in real-time. Existing systems are not
believed to provide a dynamic subset of a model which
may be user manipulated.

35

A user may choose to create and manipulate subprocesses of existing processes. In a complex environment, it may be desirable to have the ability to embed several processes in a single process for later display and manipulation. In such a manner it may be possible for users to more effectively work with complex information managements systems in a real-time setting.

10 A user normally treats business transactions as indivisible elements, each transaction making a state change to the external environment. In a complex environment, it is important to provide intermediate states for purposes of modeling that
15 environment. Present systems do not provide for such real-time modeling of business transactions. There is thus a need in the information management field for a computer system which assures that a user is working with a valid model of business transactions
20 as indivisible elements. Of particular importance in the field of complex business transactions, it would be desirable to have a system which allows a user to test different objects of a model without making irreversible changes to the preestablished model.

25

It is an object of the present invention to provide an improved information management system.

Other objects, features and advantages of
30 the invention will be apparent from the following description of the preferred embodiments thereof, and from the claims.

SUMMARY OF THE INVENTION

Briefly, the present invention is a computer system for dynamically modeling information systems
5 of organizations. The system includes a plurality of editors, or modelers, which enable a user to interact with the system. The interaction may occur via text, tree diagram, or other graphical representation. Such representations also enable the user to
10 dynamically interact with the system.

The system may include at least one information modeler, which provides a dynamic environment for the user to create, analyze, and
15 modify IM models which contain IM design data. The IM design data may be representative of entities, attributes, relationships, entity methods, partnership sets, and combinations thereof, each of which may be effected by externally applied signals.
20

An option modeler may include stored predetermined option criteria which is compared with PDL design data, or design data of the logic modeler, for syntactic permissibility. An options window may
25 then display the options which are syntactically permissible at any time during the model building process. The options window also enables the user to dynamically interact with the system.

30 The system may also include at least one process modeler, providing a dynamic environment for the user to create, analyze, and modify IFM models containing IFM design data. The IFM design data is representative of processes, information stores,

information flows, messages, sources and/or destinations. In general, the IFM design data, specifically information flow contents, correspond to IM design data entities. The IFM modeler and the IM
5 modeler can be dynamically linked, such that a change in one design data set will affect a corresponding change in the other design data set.

The system may also include a logic modeler,
10 which provides a dynamic environment for the user to create, analyze, and modify logical operations to be performed on the IM design data and the IFM design data. The logic modeler displays the system in text, tree diagram, or other graphical representations.
15 The resulting display may be dynamic, enabling the user to affect IM and IFM design data, as well as the operations to be performed on that design data.

The system can also include a function
20 modeler, which enables a user to generate a function decomposition diagram, displaying the processes of IFM design data and any related subprocesses. The subprocesses may be displayed in a hierarchical fashion, and the user may dynamically manipulate the
25 objects contained within the decomposition diagram.

The system may include filters for creating subsets of a model. Each model subset is defined by operator-determined selection criteria. The subset
30 is created by applying the selection criteria to the design data set of the model to be filtered. The resulting model subset may be displayed in a graphical representation. Such representations may

enable the user to dynamically interact with the system.

Representation of the system may be in the form of a node and pin notation, where nodes define text fields which contain statements and design data. Attached to the nodes may be pins, which identify locations where modifications may be specified for the operation associated with a particular node.

Representation of the system interfaces with the user may be established via a series of windows. These windows may include: an options facility, for displaying and selective interaction with the editors; a feedback window, for selectively displaying messages from the editors to the user; and a diagram window, for displaying representations of the system, or aspects of the system, for user interaction therewith. The system may also include syntactic and/or semantic checks of the input from the user. Errors may generate flag signals, which would appear in any of the windows.

In addition to the function modeler, a user may also "explode" a process from an information flow diagram to its corresponding subprocesses. This creates high-level processes and primitive processes. A high-level process is one which has one or more corresponding subprocesses, whereas a primitive process is one which has no corresponding subprocesses. The logic modeler may be accessed to generate a diagram from each primitive process.

In addition, the resulting exploded process may simultaneously be displayed with the information flow diagram from which it is derived. In a similar fashion, subprocesses may be "hidden" within a process on an information flow diagram or PDL display, creating high-level processes from groups of primitive processes.

The system of the invention may further include a programmed data computer adapted to establish one or more partnership sets within at least one design data, where each partnership set is representative of zero, one or more partnerships. The computer is further adapted to establish one or more partnerships, wherein each partnership is characterized by one partnership set associated with itself or associated with one other partnership set. The computer controls access to one or more design data by reference to a partnership set which is in partnership with partnership sets of one or more design data.

The system may include storage means for storing transaction data, such as entities, attributes, and relationships. The stored transaction data may then be reversibly transformed, in accordance with predetermined processing algorithms, while it is externally inaccessible. In this externally inaccessible, or static, state the transformed transaction data is evaluated against predetermined logical criteria, such as logical or semantic rules. The evaluation produces either successful or unsuccessful results. If successful, the transformed data is then made externally

-12-

accessible. If unsuccessful, the original transaction data is made externally accessible. In this manner, transaction data may be tested in an information system model without irreversibly affecting the existing model.

The system may also include storage means for storing ordered sets of references to design data, such as entities. The ordered sets, or
10 entity-sets, are assigned a value, and may then be manipulated in a manner similar to a data variable. The references may be ordered in accordance with predetermined criteria, or operator selected criteria.

15 Finally, the system may include nested blocks for iteratively processing transaction design data. The transaction design data may be representative of any of the data associated with the design data sets of any of the modelers of the
20 system, including entities, attributes, relationships, and processes, to name a few. The use of an "iterator" statement enables the system to transform transaction data at an outer subsystem and a nested subsystem level. Control is passed from the
25 outer subsystem to the nested subsystem, during operation of the system, to enable iterative processing and evaluation of transaction data against various predetermined criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a flow diagram of an embodiment of the invention.

5

Figures 1-1 through 1-72 collectively show a detailed entity-relationship diagram of the embodiment of Figure 1.

10

Figure 1A shows a diagram designating the relative position of each of Figures 1-1 through 1-72.

Figure 2 is a diagrammatic representation of display windows and facilities of the invention of Figure 1.

15

Figure 3 is a diagrammatic representation of an E-R diagram display which may be generated through an IM modeler of the invention of Figure 1.

20

Figure 3A is a diagrammatic representation of a "create" form which may be accessed through an IM modeler of the invention of Figure 1.

25

Figure 4A is a diagrammatic representation of a network diagram display of the invention of Figure 1.

Figure 4B is a diagrammatic representation of multi-layered overlapping boxes display of the invention of Figure 1.

30

Figure 5 is a diagrammatic representation of an IFM diagram display of the invention of Figure 1.

35

Figure 6A is a diagrammatic representation of an IFM diagram display of the invention of Figure 1 having exploded processes, prior to creating a new process boundary.

5

Figure 6B is a diagrammatic representation of an IFM diagram display of the invention of Figure 1 having a new process incorporating subprocesses.

10

Figure 7 is a diagrammatic representation of an FD diagram display of the invention of Figure 1.

Figure 8 is a diagrammatic representation of a PDL diagram display of the invention of Figure 1.

15

Figure 9 is a diagrammatic representation of a PDL diagram displayed in a diagram window of the invention of Figure 1.

20

Figure 10 is a diagrammatic representation of an options window of the invention of Figure 1.

Like reference characters in the respective drawn figures indicate corresponding parts.

DESCRIPTION OF THE PREFERRED EMBODIMENT

In the preferred embodiment of the invention disclosed herein, a system 10 operates using a
5 programmed IBM-compatible 80386 personal computer, having an OS/2 operating system. Other computer hardware and operating systems, such as UNIX or DOS, may be used, with some modifications to the software. Although the preferred embodiment uses C
10 as its programming language, other programming languages may be used to implement the present invention.

The system 10 assists data analysts and
15 systems analysts in building models of information systems of organizations. These models, or files, can be stored in a database and shared among analysts. In its preferred embodiment, shown in Fig. 1, the system 10 consists of four modelers, or
20 editors, which can be used to build models, each of which gives a different perspective of the same design.

The system 10 further includes a controller
25 2 and a display 3. The four modelers include: the information modeler 4; the process modeler 5; the logic modeler 6; and the function modeler 7. Each of these modelers generate corresponding models or diagrams, having design data which may be created,
30 analyzed, and modified by a user. These models and diagrams are referred to as the information model (IM), the information flow model (IFM), the Process Description Language (PDL) diagram, and the functional decomposition (FD) diagram, respectively.

Each of these models and diagrams may be dynamically linked in the manner described in further detail below.

- 5 Figures 1-1 through 1-72 collectively show a detailed entity-relationship diagram of the preferred embodiment of the invention. Figure 1A shows a diagram designating the relative position of each of Figures 1-1 through 1-72 to form a composite,
 10 complete entity-relationship diagram of the preferred embodiment.

- The following TABLE 1 identifies, in column I, elements of the system shown in Figures 1-1
 15 through 1-72. TABLE 1, column II, identifies corresponding names for certain of the elements of column I which are used in the description below.

TABLE 1

20	<u>Column I</u>	<u>Column II</u>
	ABORT	abort
	ABORT-CLOCK	
25	ALLOW-EXPRESSION	
	ALLOWABLE-VALUE	allowable.value
	ANY-SELECTION	
	APPL-ATTRIBUTE	
	APPL-ENTITY	
30	APPL-P-SET	
	APPL-PARTNERSHIP	
	APPL-STRUCTURE	information structure
	ARGUMENT	
35	ATTRIBUTE	attributes
	BEGIN	
	BETWEEN	
	BLOCK	
40	CALLED-PROCEDURE	
	CHAR-STRING	

TABLE 1 (Cont'd.)

	CHOOSE	case
	CHOOSE-BLOCK	
5	CLOCK	
	COMMIT	commit
	COMPUTE	
10	CONSTANT	
	CONTINUE	
	CYCLE	
	DELETE	delete
15	DESTINATION	destination
	DIMENSION	dimension
	DISJOIN	
	DO	do
	DOMAIN	domain
20	ELEM-ATTRIBUTE	
	ELEM-DOMAIN	
	ELEM-VALUE	
	ELSE	
25	ELSEIF	
	END	
	ENTERPRISE	
	ENTITY	entities
	ENTITY-SET	entity-set
30	ERROR	
	EVENT-OBJECT	event
	EXIT	exit
	EXPRESSION	
	EXT-CALLED-PROCEDURE	
35	FIRST	
	FLOW-ATTRIBUTE	flow-attribute
	FLOW-ENTITY	flow-entity
	FLOW-MODEL	
40	FLOW-P-SET	flow-partnership set
	FLOW-PARTNERSHIP	flow-partnership
	FLOW-STRUCTURE	message
		information flow
		contents
45	FOR-EACH	foreach
		loop
	FOR-EACH-BLOCK	
	FOR-EACH-TYPE	
50	GLOB-PROCEDURE	

TABLE 1 (Cont'd.)

	GROUP-ATTRIBUTE	
	GROUP-DOMAIN	
5	GROUP-VALUE	
	IF	if
	IF-BLOCK	
10	IM-CONDITION	
	IM-PROCEDURE	methods
	IM-PROCEDURE	entity methods
	IM-PROCEDURE	attribute methods
15	IM-PROCEDURE	partnership-set methods
	IMAGE	
	IML	
	IML-BLOCK	
	INFO-BASE	
	INFO-FLOW	
20	INFO-MODEL	information model
	INFO-STORE	information-store
	INST-APPL-ATTRIBUTE	
	INST-ELEM-ATTRIBUTE	
25	INST-GROUP-ATTRIBUTE	
	INST-SUB-ATTRIBUTE	
	INST-TOP-ATTRIBUTE	
	INT-CALLED-PROCEDURE	
	INT-PROCESS	high-level process
30	IS-ATTRIBUTE	
	IS-ENTITY	
	IS-OBJECT	
	IS-PARTNERSHIP	
	IS-P-SET	
	IS-STRUCTURE	
35	JOIN	
	KEY	key
40	LAST	
	LEAVE	skip
	LEAVE-PROCEDURE	
	LITERAL	
45	MAP	
	MODIFY	update
	MOVE	
50	NLS-CHARACTER	
	NONE	

TABLE 1 (Cont'd.)

5	NOTE NUMBER	
	ONE OPERATOR OTHER-ATTRIBUTE	
10	OTHER-ENTITY OTHER-P-SET OTHER-PARTNERSHIP OTHER-STRUCTURE	
15	P-SET	partnership-sets
	PARAMETER	
	PARTNERSHIP	relationships
	PERSON	
20	PL-ATTRIBUTE	
	PL-ENTITY	
	PL-P-SET	
	PL-PARTNERSHIP	
	PL-STRUCTURE	
25	PREDICATE	
	PROC-BLOCK	
	PROC-PROCEDURE	
	PROCESS	processes
30		process local store
	RECEIVE	receive
	RETURN	resume
	ROLLBACK	return
35	SELECT	retrieve
	SELECTION	
	SEND	send
40	SORT	signal
	SOUND	
	SOURCE	source
	STATE	
	STATEMENT	
45	STOP	
	STORE	create
	STORE	insert
	SUB-ATTRIBUTE	
50	SUB-VALUE	

TABLE 1 (Cont'd.)

	TERM-PROCESS	primitive process
	TEXT	
5	THEN	
	TOP-ATTRIBUTE	
	TOP-VALUE	
	TRANSACTION	
10	TRANSITION	
	VALUE	
	VARIABLE	
	VIEW-ATTRIBUTE	
	VIEW-ENTITY	
	VIEW-P-SET	
15	VIEW-PARTNERSHIP	
	VIEW-STRUCTURE	information flow contents
	WHEN	when
20		

The primary interface between a user and the system of the invention is through a series of interactive graphics windows presented to the user in the display 3. The windows display text, tree structure diagrams, or other graphical representations of each of the modelers. As shown in Figure 2, when a user accesses either the information modeler or process modeler, three windows will generally appear: an options facility 20; a feedback window 22; and, a diagram window 24.

The options facility 20 consists of a hierarchy of items which enable the user to perform actions on a specified model. By activating an element displayed in the facility, for example by placing and activating a cursor over the element, a selected model may be activated or deactivated. This facility may also display selections not presently available to the user at that time.

The feedback window 22 displays messages informing the user of one or more of the following: when an action has taken place; what step to take next; any mistakes or errors which may occur; when
5 the user is entering or exiting a modeler or mode.

The diagram window 24 displays diagrams generated by each of the modelers. This window 24 generally includes the capability to scroll in any
10 direction to view a diagram which is too large to fit in the window 24. Once a diagram is displayed in the diagram window 24, a user may edit the diagram, or create a new diagram. This window 24 also provides a dynamic environment for user interaction to the
15 modelers.

Additional windows may be added or included in the present system. For example, pop-up windows (not shown), context menus 26, and icons (not shown)
20 may appear at various times and at various locations on the CRT of a computer. Pop-up windows may be used to control user interactions with the system. Sometimes a pop-up window may be used to provide choices, or warn of a problem. Context menus may be
25 included to enable the user to perform actions within a specific context, or model. An item in a context menu may function in the same manner as the corresponding item in an options facility, but may occur at different times or locations than the
30 options facility. Icons may appear at various locations on the CRT, for example in the lower left corner when a model is initially opened, or as a reminder that other modelers and models are available.

Diagrams and forms of the system of the are tools used for defining and manipulating objects of the overall business information system model. Once an object is created using any of the modelers, or
5 editors, of the system, it exists in the system and can be displayed graphically in any applicable diagram, or textually in a form. When an object is manipulated, the new information about the object is accessible from any applicable diagram or form, which
10 provide different ways of accessing the same model objects by offering different perspectives of the same system.

An additional editor, the options modeler
15 5A, is available from the logic modeler 5. The options modeler is a language sensitive editor which prompts the user with grammatical options which may be inserted at any selected location. The modeler 5A defines what can be selected by the user and inserted
20 into a PDL statement, which is further described below. The modeler 5A presents the options to a user in an options window, and shows only the next available syntactic phrase or term. Thus, at any point, the option modeler provides only syntactically
25 valid options.

It is also possible to display subprocesses for a process contained in an information flow model concurrently with an information flow diagram in the
30 diagram window. In the preferred embodiment, while an activated information flow model (IFM) diagram is displayed in the diagram window, a user may selectively display subprocesses to specified process within that IFM diagram. This "process explosion" is

described in further detail in the section labelled "Process Modeler" below.

Since the four models are essentially views
5 of the same information system design, an operation applied to one model using a particular modeler has an immediate effect on another model. This may be accomplished by maintaining logical relationships between the models, thus keeping the representation
10 of design data consistent with the underlying design. It is important, therefore, that a change to a model or diagram in one modeler is reflected in other models and diagrams. For example, if a process is deleted through the function modeler, the deletion
15 is automatically and immediately reflected in the IFM diagram. By way of further example, if a process having associated information flows is created through the process modeler, the information flow model objects, or IFM design data, are reflected in
20 an activated PDL diagram. This change may occur substantially immediately, i.e. without any additional user input.

The preferred embodiment is a system
25 designed for interaction with a human user. However, the system may also be incorporated into a larger system whereby the "user" becomes a generic "operator", such as another computer. The operator may include magnetic tape devices, which download
30 information into the present system. Any other technology which permits transfer of design data, or information, such as laser disc memory devices, may be used. The operator may also be an artificial intelligence device which not only provides design

data input, but is capable of dynamically interacting with the present system.

Information Modeler

5

An information model describes the objects that are used within the system. The information modeler 4 of the present system invention includes an editor for creating, analyzing, and modifying entity relationship data models, referred to below as information model (IM), and for generating entity-relationship (E-R) diagrams which describe relationships between entities. An exemplary E-R diagram is shown in Fig. 3. The information modeler consists of IM design data, or IM objects, which may include entities, attributes, attribute methods, relationships, entity methods, keys, domains, dimensions, allowable values, and partnership sets. The modeler 4 also enables the operator to import existing information models created on other systems, or in other environments, into the present system 10.

As used in the field of the invention, the term "entity" is generally defined as an item about which information is being stored. An entity may be a tangible object, such as an employee or a part, or may be an intangible concept, such as a job title or an event. The term "attribute" is generally defined as properties of entities. For example, for an entity "customer", attributes may include address, company affiliation, or salary. A "relationship" describes the correlation among entities, or among attributes. Alternately, attributes may be "free" attributes. Free attributes are attributes not

associated with an entity. For example, an attribute may be used to hold constants, or contents of flows that go between processes. A free attribute may later become associated with an entity, or may remain
5 independent.

"Partnership sets" are described in detail in U.S. Patent No. 4,631,664 (Bachman, Dec. 23, 1986). In the present system, partnership sets are
10 used both for storing and manipulating information. Thus, using the information modeler 4, a user may define objects within the system as partnership sets. Internal to the system, partnership sets are established to provide the medium through which
15 design data is accessed and exchanged among the modelers of the present system.

An "entity method" is a PDL procedure (defined below) attached to an entity in an E-R
20 diagram rather than to a process in the IFM diagram. Each entity method defines a generic operation to be performed on all occurrences of the entity, similar to methods in object-oriented programming. The use of entity methods protects applications from changes
25 made in the design data model. Similar to an entity method, an "attribute method" is a PDL procedure associated with an attribute.

A typical use for an entity method might be
30 to specify a referential integrity rule required by a user's business. For example, a company may have a rule that requires the "last-customer-number" attribute to be incremented by one each time a new customer is created, and the assignment of that

number to the new customer. The traditional "process-centered" approach would force all developers of applications to include the code for this operation each time an application creates a new customer. However, a "create-customer" entity method may be created to handle this operation. Then, whenever an application creates a new customer, the developer calls the "create-customer" entity method which insures that the "last-customer-number" attribute is incremented by one and assigned to the new customer.

A "derived attribute" is an attribute having a value which is calculated each time it is used in a procedure. The value is not stored in the database, but is recalculated when it is used. For example, for a transaction-based system, a user might create a derived attribute, called "total-tax", that calculates the total sales tax for each sales transaction. A PDL statement is then used to create a derivation rule, which sets the derived attribute "total-tax" equal to 5% of the value of the stored attribute "subtotal-taxable." Once a derived attribute is created, it can be used in the same manner as any other attribute derived in an E-R diagram.

A key is one or more entity components that uniquely identifies an occurrence of an entity. For example, every house has a unique address and every employee has a unique employee identification number. In the preferred embodiment of the system, unique keys may represent one or more attributes, partnership sets, or a combination of both attributes

and partnership sets. Keys may be primary, alternate, or foreign. A primary key represents a combination of attributes and/or partnership sets that uniquely identify each occurrence of an entity.

- 5 For example, "Emp-Address" attribute of an Employee entity.

Any key other than a primary key is an alternate key, which may later become a primary key.

- 10 For example, a customer can be uniquely identified by a customer identification number, or by a name and address combination. A user can choose either identification form as the primary key for the Customer entity. The other key is then an alternate
15 key.

- Foreign keys represent attributes and/or partnership sets used to establish a recursive relationship or a relationship with another entity or
20 partnership set. Attributes within the entity may correspond to some or all of the attributes in the primary key of a partner entity. That is, a foreign key is a primary key in one entity that acts as a unique identifier in a partner entity, a partner
25 entity essentially being referenced by the primary key in the other entity.

- To create a new information model, the user selectively interacts with the options facility 20 to
30 the information modeler 4 to indicate that a new file is being created. The options facility 20 may include various forms of information display, such as a palette, a dialog box, or context menus, depending upon the type of information to be displayed and/or

ergonomic criteria. The new information model (IM) is named, and identified as to type of model. In the preferred embodiment, a diagram is generated in the diagram window 24. An operator may then create an IM 5 by creating and placing IM design data, or objects, in the IM diagram. The IM design data may include entities, with associated attributes, relationships, text blocks, and entity methods. In the preferred embodiment, IM design data may include at least the 10 following objects and combinations: attributes; attribute methods; entities; entity methods; partnership sets; partnership set methods; entities and relationships; entities and attributes; and entities, relationships, and partnership sets.

15

To create one or more entities, an operator selects an element in the context menu 26 corresponding to "create". A "create" form 40, shown in Figure 3A, is then displayed in the diagram window 20 24. The operator then describes the entity by entering the following information: entity name; description; name of parent entity; and expected volume. This information is then stored, and selectively displayed upon later recall by either the 25 operator or one of the modelers.

Attributes may be created in a manner similar to entities, and are described by the following information entered by an operator: 30 attribute name; description; expected length; domain; data type. Data type of an attribute generally may include integers, strings, and the like. Domains may be used to define useful, recurring data types. For example, "date" is a domain, made up of separate

pieces of data, each of which may be defined as subdomains. Attributes may only be assigned superior domains as data types. A domain can consist of one or more pieces of data. In the present system, deleting a domain does not delete attributes with that data type, but does leave attributes without data types. Dimension allows a user to group domains which are comparable in value. For example, a value space may have one or more representations such as U.S. format (e.g., November 2, 1990), or European format (e.g., 2 November 1990). By making a dimension "date", all attributes having "date" data type, regardless of format, will be shown.

15 In the preferred system, attributes may be created independent of entities. Using the partnership set protocols described in U.S. Patent No. 4,631,664, relationships may be one-to-one, one-to-many, or many-to-many, depending upon the type and complexity of the system or model created.

Once an IM is created, it may be displayed as an IM diagram, as either network diagrams or multi-layered overlapping boxes, shown in Figures 4A and 4B. Other graphical representations may be used, such as text or tree structure diagrams (not shown). Each design data, or object, of the diagram, or text, may be dynamically activated by the user to modify, analyze, or create the associated IM. Thus, the information modeler 4 may be accessed by the operator by means of the diagram window 24. In a similar manner, the IM diagram 30 may be manipulated as to size or object location. In the preferred embodiment, a pointing device, such as a mouse, is

used in conjunction with the system 10 to manipulate a pointing icon in the related windows. Thus, the pointing icon may be used to expand or contract the size of the diagram, the size of the design data associated with the diagram, and placement of the design data in relation to the diagram.

Process Modeler

10 The process modeler 5 of the preferred embodiment is an editor used to define an information flow model (IFM). An IFM describes a function that occurs within the system. The process modeler 5 enables an operator to define the processes,
15 information stores, information flows, messages, and sources/destinations of the system. The information store of a process modeler 4 corresponds to design data elements in an IM. The information flow of a process modeler represents the flow of design data
20 elements which are usually defined in an IM. Therefore, in the preferred system 10, the IM is generally established before the related IFM. The IM to which an IFM is attached can be empty; that is, it does not have to contain entities, attributes, or
25 other elements. Thus, an operator may begin modeling a system with an IFM diagram, generated through the process modeler to describe relationships between objects in a process model. An exemplary IFM diagram
50 is shown in Figure 5.

30

A new IFM can be created through the process modeler 5 in the same manner that a new IM can be generated through the information modeler 4, as discussed above. Generally, creating a new IFM

requires operator input of the following: a model name; identification of an associated information model; target database in which the IFM will be stored; and model type. An IFM diagram may be associated with the process modeler 5 such that a user may dynamically interact with the diagram in creating, analyzing and modifying an IFM. Once an IFM diagram is generated, after the above-mentioned information is entered by the user, IFM design data, or IFM objects, may be selectively placed on the IFM diagram. The IFM design data includes processes, information flows, messages, sources/destinations, process local stores, events, flow-partnership, flow-partnership set, flow-entity, flow-attribute, and information stores. After one or more IFM design data are created by the operator, information flow can be added between them. Duplicate names for design data or flows are generally not accepted within an IFM.

20

In the preferred embodiment, an information flow may be added to the model once at least two processes, or a process and an information store, or a source/destination object are present. Generally, data cannot flow between two information stores, or between two source/destination objects without an intervening process. Therefore, the preferred system accesses information flows created between two processes, between a process and a source/destination, or between a process and an information store. The contents of an information flow are generally referred to as messages, which may consist of design data.

30

Information stores may be created to contain one or more entities, and may be included in the process model. There is an association between information stores and entities, which may be graphically represented in E-R diagrams. An entity may exist in one or many information stores. Therefore, the IFM diagram, IM diagram, and associated process model and information model forms are linked by the association of the information store and the entity. Forms, like diagrams of the preferred system, may be used as tools for defining and manipulating all objects of the system. Diagrams and forms provide graphical and textual ways of accessing the same objects of the system.

15

In the preferred embodiment, Events are the one control flow design data in the IFM diagram. Examples of Events include Clock/Calendar Events, User Events, Database Events, and Flow Events. A Clock/Calendar Event may indicate that a process is to execute at a specified time. For example, a user might specify a process to run every day at 2:00pm. User Events generally correspond to a user demand, for example, an explicit "run this process". A Database Event may be a mechanism that initiates the execution of a process based on the changing of an information store value. For example, if a customer goes from having good credit to having bad credit, this might trigger the "send threatening letter" process. A process triggered by a Flow Event begins execution upon the arrival of the flow structure. Flow Events must correspond to a flow going to that process, or a triggering flow.

30

While events generally appear in IFM diagrams, there may be corresponding clauses in other modelers of the system. For example, in the preferred embodiment, a Triggered-When clause in the logic modeler corresponds to the IFM diagram Event for that process. Further, in the preferred embodiment, every completely specified primitive process (described further in the "Function Modeler" section, below) must have a triggering Event associated with it. This Event is what actuates the process, forcing it to begin "executing" its business logic.

Source/destination objects may be defined to direct information in an IFM. In the preferred system, these objects may be included in the IFM diagram to indicate the source or destination of IFM design data of that diagram.

The size of an IFM diagram may be changed, enlarged or diminished, to accommodate the viewing needs of a user. In the preferred embodiment, a user can generate an expanded view of a selected process or design data element. The size of the IFM diagram may also be changed to accommodate an overlapping, or superimposed exploded process or function decomposition diagram (described below). In addition, the layout of the IFM diagram may be modified by moving IFM design data, for example, boxes and graphical representations (arrows) of information flows connecting the boxes, as shown in Fig. 5.

Once a process is created it may be divided into lower-level processes or subprocesses. Dividing a high-level process into two or more subprocesses is referred to as "explosion". In the preferred embodiment, the process to be exploded must have at least one information flow attached to it, either incoming or outgoing. In other forms, such a restriction is not necessary. When a process is exploded, the number of subprocesses generated and named by the system for the exploded process equals the number of flows coming into and out of the process.

In one embodiment, a subprocess 58 related to an incoming flow has the word "handle" as a prefix to its name; a subprocess 60 related to an outgoing flow has the word "generate" in its name. Subprocesses 62 may exist within a process 56 which have neither incoming nor outgoing flow associated therewith. For example, as shown in Figure 5, a process "fill orders" may be divided into subprocesses "generate inventory update", "handle orders for shipping", "put orders in boxes", and "take boxes to mail room".

Thus, an IFM diagram may contain both high-level processes 56 and primitive processes 62. High-level processes 56 are those processes having subprocesses associated therewith. Primitive processes 62 are those processes having no subprocesses. Once a process is exploded, the subprocesses may be displayed through a functional decomposition diagram 80, (defined below), or superimposed over a segment of the IFM diagram.

In the preferred embodiment, a user can simultaneously and concurrently view an IFM diagram 50 and an exploded process 56 in the same diagram window 24. For example, a user may use a pointing device to point to an icon representative of a high-level process 56 in an IFM diagram 50, and display the "explosion", or subprocesses, of that process, while maintaining the other processes on the display screen. A user may also display or hide the subprocesses in a corner window, leaving the IFM diagram intact. Subsequent changes to an IFM object or process in either window would be reflected in the other window.

15 In an IFM diagram 50 a user can create a high-level process 56 for a process or group of processes by defining process boundaries 70. In the illustrated embodiment, and as shown in Figures 6A and 6B, the system prompts the user to name the new high-level process, and then draws the new process 74 around the selected, or "boundaried", processes 72. The original processes 72 become subprocesses of the new process 74. Making a new process boundary 70 is a way of changing the hierarchy of processes in the process model. Each new high-level process has a form. This form lists the subprocesses in the process hierarchy fields of display. Likewise, the forms for each subprocess list the new process as the top-level process. In a similar manner, process boundaries 70 may be undone, returning the process model to the state it was in before the process boundary 70 was created.

Function Modeler

Once an IFM containing process is created, the function modeler 7 may be invoked to generate and display a function decomposition (FD) diagram of the process, as exemplified by diagram 80 shown in Figure 7. The decomposition diagram 80 consists of subprocesses 84 identified by the operator to be associated with an identified process 82. For example, as shown in Figure 7, if the process 82 is identified as "fill orders", subprocesses 84 identified with that process 82 may include, "generate inventory update", "handle orders for shipping", "put orders in boxes", and "take boxes to mail room". In the preferred embodiment, a pointing device, such as a mouse, is used to select a process displayed in an IFM diagram. By selecting a process with the pointing device, a decomposition diagram may be superimposed over all or part of the IFM diagram.

20

A typical FD diagram 80 of the present invention is shown in Figure 7. There, the diagram 80 is a tree-structure type diagram, having nodes containing process description 86, process 82, and subprocess 84. The relationship between the nodes is shown graphically by a main line 88, and branch lines 90. The nodes are each connected to the lines 88, 90 at root pins 92. Each of lines 88, 90 also has an extension pin, 94 at which nodes may be attached.

30

An important aspect of the present invention is that the FD objects of the FD diagram 80 are dynamically linked to the IM and IFM objects such that a change to the FD diagram 80 (Fig 7) is

substantially immediately reflected in the IFM diagram 50 (Fig. 5) and the design data sets of both the information modeler and the process modeler.

5 In addition, a FD diagram 80 may display subprocesses 62 created in a process explosion. Thus, there are two ways in which subprocesses 62 may be created and displayed: using the process explosion from an IFM diagram 50; or using the
10 function modeler to create a FD diagram 80.

Logic Modeler

The system 10 of the invention also includes
15 a logic modeler 6 which specifies the logic of primitive processes defined in an IFM. The logic modeler 6 primarily embodies a graphical, structured language which enables a user to define business logic of the processes that operate on entities
20 defined in an IM. The logic modeler 6 may be used to define logic models for describing the flow of information between models.

The logic modeler 6 refers to entities and
25 attributes in the IM as it defines the logic that operates on the entities and their attributes according to the organization's business rules. For example, the business rule for deriving an attribute can be expressed in the logic modeler 6 and stored.
30

In the preferred embodiment, the structured language is a Process Description Language (PDL), which shows the flow of processing in relation to objects in the IFM and the specific operations

performed on entities and attributes in the information model. The logic modeler 6 is generally related to an IFM and to an IM. As shown in Figure 8, a PDL diagram 100 may be built by creating PDL statements, one at a time, in a window. The system then automatically generates graphical representation of the statement in the PDL diagram 100. The logic modeler 6 includes PDL procedures and PDL statements in describing and displaying logical operations on objects. As discussed above, entity methods and derived attributes may be used to define a procedure on an entity in the corresponding IFM diagram 50. PDL procedures may also include process procedures, which describe the business logic of the primitive process, and derived attributes, which are attributes having a value calculated each time it is used.

PDL statements may be used to build procedures and write specifications that describe the business logic of primitive processes in a corresponding IFM diagram. Statements may be built by a user or generated by the system, using keywords, identifiers, and expressions. Keywords which may be used in building PDL statements may include, inter alia: CREATE; RETRIEVE; UPDATE; DELETE; SEND; RECEIVE; SIGNAL; and, WHEN.

In the system of the present invention, several types of statements are available, including logical access statements (RETRIEVE, CREATE, UPDATE, DELETE), iteration statements (FOREACH, and LOOP), conditional statements (IF, CASE), and control transfer statements (EXIT, SKIP, RESUME). Logical access statements describe selecting or changing data

defined in information stores. These statements are equivalent to statements that describe database access in a physical database. Any of the logical access statements of the present embodiment can
5 create new objects in the IM diagram and in the IFM diagram.

In the preferred embodiment, "send" and "receive" statements are also available. The "send"
10 statement defines the destination of information leaving a particular process. It can create an information flow from the process a user is currently editing to another process or source/destination. The syntax, or semantic rule, associated with a
15 "send" statement is:
SEND-(INFORMATION-STRUCTURE-REFERENCE)-WHERE
(optional) - THROUGH (optional) - TO (optional).
"Information-structure-reference" specifies the information to be sent from the process. The
20 information can be a list of attribute references or an "entity.*" (indicating all attributes of a particular entity). An attribute reference can be an attribute name or a qualified attribute name. Optional clauses may include "where", "through", and
25 "to". The "where" clause specifies search conditions that determine the information to send. The "through" clause specifies the information flow through which a user wants to send information. In one embodiment, only one information flow is
30 allowed. The "to" clause specifies one or more processes or source/destination objects to which a user wants to send information.

The "receive" statement specifies the source of the information entering the process. It can create an information flow from a specified process or source/destination to the process for which a user is currently specifying PDL. The syntax associated with a "receive" statement is: RECEIVE - (INFORMATION-STRUCTURE-REFERENCE) - WHERE (optional) - THROUGH (optional) - FROM (optional). The "information-structure-reference" is the same as for "send" statements. Similarly, the "where" clause specifies search conditions that determine the information to receive, the "through" clause specifies the information flow through which a user wants to receive the information, and the "from" clause specifies the process or source/destination object from which the information is to be received. In one embodiment, neither the "send" nor the "receive" statements are allowed in entity method and derived attribute procedures. Both SEND and RECEIVE can create entities and attributes in the information model by reference. That is, if a user references an entity or attribute which does not exist, the system will prompt the user to create such an entity or attribute. Thus, similar to logical access statements, SEND and RECEIVE statements of the present embodiment can create new objects in the IM diagram and in the IFM diagram.

A SIGNAL statement may be used to signal an exception condition or error and to transfer control to a condition handler (WHEN clause) defined for the specified condition. A syntax for the statement is: SIGNAL - condition - (- expression list -). The "condition" is one of the system-defined conditions,

including: ABORT, DUPLICATE, EMPTY, ONE, BEGIN, FIRST, LAST, EXIT, BETWEEN, END, END GROUP, SKIP, and, OTHERWISE. The system may test for and signal some of these conditions by default. An

5 "expression-list" is a user-defined list of parameters to be passed to the condition handler.

A condition handler, such as WHEN, may be included to specify the action to take when a

10 particular condition arises. This may be used in place of a user-written PDL statement to test for or signal that a condition exists. This may be used in conjunction with a SIGNAL statement to explicitly signal any of the system default conditions. Thus,

15 as an argument to the SIGNAL statement, a user may specify the condition to be signaled and any parameters to be passed to the condition handler.

Two other important PDL statements are

20 CONNECT and DISCONNECT. In general, these statements provide the means by which two instances of entities are caused to be joined at one of their partnership sets by a single partnership. The CONNECT statement may provide the means for causing a new instance of a

25 partnership to be created joining two instances of a partnership set, each of which is associated with an instance of an entity. In the case of non-directed, recursive partnerships, the two ends of a single instance of a partnership must both be attached to

30 the same instance of a partnership set.

In the preferred embodiment, when a CONNECT statement is executed, it is necessary to establish two sets of information; one set for each end of the

partnership being created. In that instance, each set must include: (1) identification of the type of entity and instance to be connected; (2) identification of the type of partnership set to be connected; and, (3) identification of the rules of insertion to be applied to the new partnership endpoint. These rules may include identification of the new partnership endpoint as the "first" member, as the "last" member, as the "prior" member to some designated current member, as the "next" member after some designated current member, sorted by some attribute value controlled by sorting rules, or in no particular order. Where the maximum cardinality is one, the rules of insertion are moot, as there can then be only one instance of a partnership endpoint at an instance of a partnership set at any moment in time.

The CONNECT statement may be an independent statement, or may be included as an extension of an INSERT statement. In the preferred embodiment, where the minimum cardinality is one (1) for one of the partnership sets, then the CONNECT statement must be treated as an extension of the INSERT statement to assure that the integrity of the partnership set is maintained after the INSERT is completed. In the situation where the minimum cardinality is one (1) for both partnership sets of the partnership, it may be necessary to actually INSERT both entities and CONNECT them before the information structure is in a consistent state. Where the minimum cardinality is zero for both ends of the partnership, or where the maximum cardinality is greater than one (1) for both ends of the partnership, it is essential that there

exist an independent CONNECT statement which may add partnerships as they are required.

The DISCONNECT statement provides the means for causing an existing partnership to be destroyed. The destroyed partnership might be replaced by a new one, several, or none, depending upon the information situation.

- 10 Where it is essential to the integrity of the data definitions, it may be necessary that a RECONNECT statement be constructed as a separate command, or as a pairing of a DISCONNECT statement and a CONNECT statement to assure that the minimum
- 15 cardinality rules remains intact. Where the order of the members of a partnership set is defined as being controlled by the values of one or more attributes, then the modification of the value of one of those attributes might require that the members of a
- 20 partnership set be reordered to maintain the integrity of the declarations. The SORT statement may be defined to cause the members of a partnership set to be placed in a defined sequence on a batch basis, rather than on an ad hoc basis as they are
- 25 created.

Other statements available in the illustrated system, plus a brief description, are listed below in TABLE 2.

TABLE 2

	ABORT	signals fatal error and rolls back all changes specified by the transaction
5		
	CASE	executes zero, one or more of several possible statement lists; compares several possible values to the value of "case selector"
10		
	COMMIT	transfers process results to information store
	DO	calls an entity method or procedure from a process procedure, another entity method, or the same entity method
15		
	EXIT	used in iteration blocks to terminate the iteration statement and transfer control to the EXIT condition handler or to the statement following the iteration block
20		
	FOREACH	starts an iteration block, processing a single entity occurrence on each iteration until a specified condition is met
25		
	IF	makes the execution of a block statement conditional
30		

TABLE 2 (Cont'd.)

5	LOOP	starts an iteration block, processing the statements within the block until a specified condition is met
10	NOTE RESUME	adds comments to a PDL specification used in condition handlers to indicate that the condition handler completed successfully, and transfers control to the statement following the SIGNAL statement that raised the condition
15	RETURN	exits from the process with a value
20	SKIP	terminates the current iteration and transfers control to the SKIP condition handler or to the next iteration of the block

Other embodiments may have other statements which perform substantially the same functions as the CREATE, RETRIEVE, UPDATE, DELETE, SEND, RECEIVE, or other statements of the preferred embodiment described above. The description above is not intended to be semantically limiting, but merely to describe functions for affecting the modelers and associated design data.

30 The logic modeler 6 may be accessed from any of the other three modelers. In general, an IFM having the processes upon which the logic modeler 6 is to act must be opened prior to accessing the logic modeler. When a user creates or modifies objects in

the IFM diagram 50, the system 10 automatically creates or modifies the corresponding PDL statements and their graphical representations. For example, if an information flow is created from a

5 source/destination object to a primitive process and the contents of the information flow in the IFM diagram 50 are defined, the PDL diagram 80 for that primitive process contains a node labeled "receive" (nodes are described in detail below). The "receive"

10 node represents a PDL statement that describes the source of information coming into the process.

In a similar manner, when PDL objects are created or modified, the system creates or modifies

15 the corresponding objects in the IFM diagram 50. For example, any of the PDL logical access statements can create a new entity in the IM diagram 50 or a new information store in the IFM.

20 The PDL diagram 100 may also include graphical representation of the process description, and may enable a user to generate a report containing the text of a process description. In the preferred embodiment of the system 10, the graphical

25 representation is in the form of a tree structure diagram, shown in Figure 8. The tree 100 may have a series of nodes 102, pins 104, and PDL statements 106. The nodes 102 appear as boxes on the tree structure diagram 100. The nodes 102 contain text

30 fields, upon which statements from the modelers or from internal language statements, appear.

Pins 104 represent the relationship, or link, between statements or parts of statements. The

position of a pin 104 in relation to its node 102 indicates the type of pin 104 it is. As best shown in Figure 9, pin types may include: extension pins 112; root pins 114; parent/child pins 116; qualifier pins 118; and condition pins 120. An extension pin 112 may be used to mark each location where an operator can add nodes 102 to the end of an existing sequence. An extension pin 112 may appear at the end of a main line 108, or at the end of each line that represents a subsection of a process description. A root pin 114 may be used to mark the location, or root, of a node 102. Selection of a root pin 114 by an operator enables the operator to insert nodes 102 before the root pin 114. A root pin 114 may appear along any line of a process description. A parent/child pin 116 may be used to mark the beginning of a subtree of nodes, and allows a user to insert nodes 102 at the beginning of a subtree. A qualifier pin 118 may be used to represent a clause that modifies a PDL statement 106 contained in an associated node 102. A condition pin 120 may be used to represent directions for handling exception conditions and errors associated with statements in a node 102. An important aspect of the nodes/pins relationship is that each pin 104 is associated with at least one node 102, and is graphically represented as such.

In general, each node 102 and its attached pins 104 represent a PDL statement 106. Pins 104 may also have active or inactive, and may be displayed to the user independent of its status. A pin 104 having active status is defined as a pin 104 which acts upon the associated node 102, whereas a pin 104 having

inactive status is defined as a pin 104 which merely represents the possibility of becoming active upon user input.

- 5 The logic modeler 6 may have a series of windows associated with it which are unique to the logic modeler. These windows may be similar to those used in the other modelers, but may only reflect activities within the logic modeler and associated
- 10 logic model.

- PDL statements and their graphical representations can be created either by being generated by the system 10 from objects in the
- 15 corresponding IM and IFM, or entered directly by the operator. The graphical representations may be in the form of a tree structure diagram 100, a network diagram (not shown), or other forms customary in the art, which may be displayed in a diagram window 24.
- 20 In the preferred embodiment, a node 102 representing the scope of the description occurs at the top of each PDL diagram 100. The node is generated by the system 10 to describe the process for which the user is specifying a logic modeler 4. All possible
- 25 qualifier pins 118 and a condition pin 120 appear on the node 102 independent of whether a user has actually entered them as part of the statement. In the preferred embodiment, if a qualifier pin 118 is active, i.e. actually part of the statement, the text
- 30 of the clause represented by the pin 118 appears to the side of the pin 118. If a pin 103 is not active, it serves as a place holder, telling the user that a clause may be added to the statement.

Addition or modification of design data, or objects, in the PDL diagram 100 may be achieved by establishing an editing focus, which may be user-selected and system activated. An editing focus may be a pin 104 to which the user wants to add a node 102, or it can be a node 102 or pin 104 the user wants to modify. When a new node 102 is added to the end of a LM tree of a PDL diagram, the logic modeler automatically creates a new pin 112 that extends from the new node. This extension pin 112 then becomes the editing focus. An operator can easily change this editing focus by selecting another object in the displayed PDL diagram 100.

15 In the preferred system, the logic modeler may also include an options modeler and related options windows which present items available for the operator to use in building PDL statements 106 and manipulating PDL diagrams 100. An exemplary option window 150 is shown in Figure 10. The options modeler is a language sensitive editor which generates a display of syntactically valid options for insertion by the user at a selected point in a line of a procedure, specification, or statement. 25 The options window shows only those terms and/or phrases which are syntactically valid for building PDL statements, and thus may be displayed as a PDL palette as shown in Figure 10.

30 The syntactic validity of an option is determined by comparing the options available, to criteria which may be defined by logical relationships, general semantic rules, expert rules, and heuristics. Logical relationships may include

requirements such as certain keywords must be used in conjunction with specific identifiers. General semantic rules may include requirements that two keywords cannot occur next to each other without 5 intervening identifiers. Expert rules and heuristics may involve series of user-defined logical correlations which must occur before a PDL statement structure is considered correct.

- 10 In the illustrated embodiment, the options window 150 is a tree-like display of items that a user can use to build PDL statements. These items may include keywords, symbols, identifiers, and the names of objects in the corresponding IM diagram 30 15 and IFM diagram 50. When the logic modeler 6 is first displayed, a single PDL palette 152 appears in the options window. Once a user selects an editing focus, the PDL palette item expands to display one or more categories of the PDL grammar. For example, 20 when a user sets the editing focus to an extension pin 112, a root pin 114, or a parent/child pin 116, the following items appear in the options window: verbs, control flow, qualifiers, NOTE. The first three items are categories of the PDL grammar. The 25 fourth item is a PDL keyword that can be used to add comments to procedures.

In other systems, an option modeler may be used in conjunction with any of the modelers, or 30 editors, in designing and building a model.

Additional Aspects of the System(1) Dynamic Linking

5 An important aspect of the present system is a dynamic linking between the design data sets of each modeler. Within the logic modeler, when design data is created or modified in the information modeler or process modeler, the system automatically
10 creates or modifies the corresponding PDL statements and their graphical representations. For example, if an operator creates a source/destination design data, or object, with an information flow to a primitive process in the IFM, the PDL diagram 100 for that
15 primitive process contains a node 102 labeled RECEIVE. The RECEIVE node represents a PDL statement that describes the source of information coming into the process. In a similar way, when an operator creates or modifies PDL objects or design data in the
20 logic modeler, the system creates or modifies the corresponding design data in the IFM and/or IM.

 The dynamic characteristics of the system of the invention may be achieved by use of partnership
25 sets. As best described in U.S. Patent Number 4,631,664 (Bachman, 1986), partnership sets may be used to control storage, retrieval, and modification of information in a design data collection contained in storage devices in a design data processing
30 system. In the present system, partnership sets are used to handle complex information sets, as described above.

(2) Dynamic Text and Graphics

In addition, each of the modeler displays may include text having a combination of icons and 5 alphanumerics. The text may be dynamic, such that a user may affect IM design data, IFM design data, FD design data, and PDL design data by interacting with either the icons or alphanumerics. Since the present system is a primarily graphics system enabling an 10 information system user to create graphic models of complex information systems, the ability to use dynamic icons and alphanumerics may be an important feature of such a system.

15 (3) Diagnostics

Further, the system of the invention may include an internal diagnostic or consistency check which signals to a user when improper terminology is 20 used. Certain processes, semantic rules, logical relationships, expert rules, heuristics, text fields and pins may be stored in a database storage. When a user creates, modifies or adds design data in any of the modelers, the present system will check those 25 statements against internally-stored design data. If the input design data does not correspond to the predetermined design data, the improper design data is "flagged" by the system and appropriate flag signals are displayed to the user.

30

In addition to being a double-check system for syntactic and semantic correctness of statements, the signals may be a part of an iterative process performed within the system. In the preferred

embodiment, a design task may be included which is an intelligent, rule-based task that automates the repetitive process of examining and modifying the information model. The design task may include expert rules and heuristics against which statements and procedures are checked.

In the illustrated embodiment, the design task works in the following series of steps: (1) using a built-in set of design rules, the design task searches the information model for objects that might be more complex than necessary; (2) it then displays a list of those objects and actions that might simplify those objects; (3) the user then decides what action to take next, which may be either selecting an action that modifies one or more of the objects, or marking one or more of the objects as correct, thus indicating that the rule is inappropriate for the particular object; and, (4) the user then decides whether to rerun the same design rule, or to run the next rule.

(4) Filters

A user may also define and implement filters. A filter allows a user to view and operate on a subset of a model. The objects that are included in a subset, or filtered model, are determined by selection criteria defined by a user when the filter is created. In the preferred embodiment, multiple filters may be created for the same model, however only one filter can be active at a time. For example, if a user wants to display objects created by a specific user and also display

objects of a certain type, two filters can be defined, each including those respective objects. Both of these filters would be applied to the same original model, but they would each produce a subset of that model.

When using the filters, if a diagram is displayed when a filter is activated, the filter takes effect immediately. That is, all objects which do not meet the selection criteria of the filter are automatically cleared from that diagram. It may be possible to implicitly include objects when an object is explicitly selected to be included in a filtered model. For example, if an information flow is the selection criteria of the filter, the objects that are the source and destination of that flow are implicitly selected, but not associated subordinate objects, such as flow structures. Similarly, when a diagram of a filtered model is displayed, objects that meet the filter selection criteria are displayed, along with any objects upon which their existence may depend.

(5) Process Description Language (PDL)

25

The system of the present invention enables a user to describe the elements and concepts most important to a business organization. Building a model of complex organizations facilitates the communication of vital business functions among the many departments of an organization. In the preferred embodiment, PDL enables a user, in a specification/design mode, to specify business logic for a model. Objects and processes (i.e. PDL design

data) are specified and/or defined. In an execution mode, PDL enables a user to manipulate instances of objects and processes of the model. Such objects and processes, in both modes, include entities, attributes, processes, messages, and relationships. In both modes, the processes may be independent processes, or cooperating processes which are focused upon the successful processing of individual "business transactions".

10

The transactions specified by PDL are business transactions which are declared to be indivisible elements. The PDL essentially enables the system to assure that a user is working with such indivisible elements by providing a means for defining pieces of discrete business processing. The business world, or environment of an organization, can thus be defined in terms of discrete transactions which are known and either processed or not processed. In a model for complex organizations defined in terms of business transactions, or indivisible elements, validity checks need to be performed each time an instance of an object or process is manipulated or a process is defined. To do so, user data is effectively maintained in a static form until the specific transaction either becomes "successfully" or "not successfully" processed.

30

The business transactions related to the processes specified by the PDL are thus indivisible elements of work which are to be treated by the business as being in one of four states: waiting to be processed; in process; completely and successfully

processed; or, completely and unsuccessfully processed. In the preferred embodiment, the rules expressed by the PDL for one business transaction can only either specify access to information in an information store that has been placed there by completed other business transactions, or specify receipt of messages from completed business transactions. During its processing period, i.e., in the execution mode, the PDL enables the storage of new information in the information store, and the sending of information intended for other processes. Moreover, the stored information will immediately be available to processes cooperating on the same business transaction. In the case of an unsuccessful process, it may make changes to the user data, and send messages reporting an unsuccessful transactions after its prior actions have been undone. Processes working on other business transactions will not be able to access the new information, nor will the processes be aware of the generation of the messages until after the source process has either successfully terminated or declared that it has been aborted.

25 Thus, in execution mode, the present system with PDL enables a user to manipulate instances of objects and processes within a model. That is, the user data is essentially put into a static state during which operations are, in effect, reversibly performed without affecting existing instances of
30 objects or processes. That is, changes to the user data are performed in "quarantine", and are not incorporated into the user data until the process is determined to be (in accordance with criteria or

algorithms defined by the PDL design data), and reported successful or unsuccessful. A user-specified process may then abort the action without side-effects to the existing user data, or
5 commit the action. This is analogous to having a master "undo" command which retrieves data deleted, but temporarily stored, in an application. The changed user data may be made accessible for processing in conjunction with other business
10 transactions.

In the preferred embodiment, a PDL ITERATOR statement may be a major block of a program, having a number of nested blocks. Each nested block contains
15 the logic to be executed under a designated circumstance or set of circumstances.

(6) Entity-sets

20 The system of the preferred embodiment may also include entity-sets. In the specification mode, entity-sets are defined in terms of references to entity types and conditions. In the execution mode, an instance of an entity-set contains references to a
25 list of entity instances, which are then treated as a value. Thus, in the execution mode, entity-sets may be assigned to a variable or to the attribute of an entity, or may be used as an operand. For example, such entity-sets may be the operand of an ITERATOR
30 statement.

An entity-set variable provides the means for storing and accessing a list of references to entity instances. In the execution mode, this

collection of references is ordered, i.e., there is a "first" reference, a sequence of "next" references, and finally a "last" reference. The sequence of references is preserved, unless some explicit action is made to reorder that set. New references may be added to, deleted from, or rearranged in the list. It may be possible for one entity instance to be referenced in an entity-set more than once, however explicit action may be used to hide duplicate references to a single entity instance. The type of entity instances referenced within an entity-set may be of the same or different entities.

An entity-set is a generalization of the partnership-set of the partnership data model. It takes the properties of a partnership-set expressed as a particular property of an entity and characterizes it as a holder of a list of entity references. These properties are transferred to a new type of data variable, which is not necessarily identified with a particular entity. The assignment of a list of references from a partnership-set has the effect of transforming that list from a dynamic list, responsive to the creation and deletion of partnerships between the subject partnership-set and other partnership-sets, into a static list subject only to explicit commands addressing that list.

Generally, data variables are named storage structures, where each data variable is specialized to handle particular types of data. In the present embodiment, during operation in the specification mode, an operator may declare a variable to be an entity-set variable; during operation in the

execution mode, a user data value may be assigned to an entity-set variable. Thus, this leads to a new type of data variable, i.e., one that can store and return the values which are references to entity instances. In the preferred embodiment, an entity-set has three basic states: empty (i.e., the entity-set has no members); not-empty (i.e., the entity-set has a list of one or more entity references); and, undefined (i.e., no entity-set, either empty or not empty, has been assigned to the data variable). The undefined state may appear initially after the data variable has been established but before any value has been assigned to it. The undefined state may also arise subsequent to a data variable being nullified.

The association of data variable with a data type entity-set enables the separation of the idea of the set of entity instances associated in partnership with a partnership-set, from the act of iterating over the list of entity instances of a partnership-set to perform the actual processing of those entity instances. In the preferred embodiment, a SELECT statement may be used to assign an entity-set to a data variable, and to hold that entity-set value until it becomes appropriate to take some action against the set of entity instances referenced by the entity-set value.

Establishment of entity-set data types enables the identification of data variables, including those which are process-local variables, as well as those attributes of entity instances which may be defined, as having the data type of

- entity-set. Further, any operation which may be performed upon an entity instance may also iteratively operate on a set of entity instances by referencing a variable of the type entity-set.
- 5 Specifically, within the logic modeler of the preferred embodiment, PDL statements may act upon entity-set variables. Operations may also involve two or more entity-sets. For example, entity-sets may be merged, concatenated, compared, joined, and
- 10 tested for the inclusion of one within the other, or the exclusion of one from the other.

- While, in the preferred embodiment, PDL is used to manipulate business transaction design data,
- 15 other embodiments may use mechanisms independent of the PDL of the present system. Business transaction design data may correspond to transaction-related information, such as rules of transactions which operate on entities, attributes, and relationships.
- 20 These rules may include processing algorithms, and may be stored internal to a computer system. User data may then be transformed in accordance with the transaction design data.

- 25 The user data of the entire business transaction model described above may be held in an externally perceived static state, while that data is effectively internally transformed in a reversible manner. The discrete elements of the user data of a
- 30 business transaction may be tested against the existing user data without making externally perceived changes to the existing user data. Once the internal check is complete, in accordance with the predetermined algorithms, external signals may be

sent by a user to either commit (if "successful") or abort (if "unsuccessful") the changes to the user data. Subsequent to the abort signal and the reversal of prior action on the user data and
5 messages, the process may make changes to the user data and send other messages that are relevant to the unsuccessful transaction (e.g. indicative of the unsuccessful processing). The changed user data then becomes fully externally accessible.

10

The specification, or outcome, of the system described above, may be converted into code and implemented to produce a working application system. That conversion may be accomplished manually, or
15 under the control of a programmed digital computer. This specification, or outcome, is preferably in terms of PDL statements similar to those discussed above and as listed in Table 2.

20

Appendix A, attached hereto, is a complete printout of the object code for the system described in detail above and is incorporated herein by reference.

25

Appendix B, also attached, is a manual describing a preferred embodiment of the present system.

30

The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiments are therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the

appended claims rather than by the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

Claims:

1. A computer system for establishing a dynamic model of information systems of organizations comprising:
 - 5 A. at least one information modeler including IM means for creating, analyzing and modifying an information model, said information model having IM design data representative of elements including at least one from the group
10 consisting of attribute, entity method, attribute method, partnership set, and partnership set method, said attribute being independent of an entity;
 - B. at least one process modeler including
15 IFM means for creating, analyzing, and modifying an information flow model, said information flow model having IFM design data representative of elements including at least one from the group consisting of processes, information stores, information flows,
20 messages, views, process local stores, sources, and destinations, said IFM design data relating to at least one of said elements of said IM design data;
 - C. modeler linking means for dynamically
25 correlating said information modeler and said process modeler, whereby a change to any of said IM design data affects a corresponding IM-IFM change in said IFM design data, and whereby a change to any of said IFM design data affects a corresponding IFM-IM change
30 in said IM design data; and

D. operator means for applying external signals to said information modeler and said process modeler, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said IM means and said IFM means.

2. A computer system according to claim 1 wherein said elements of said IM design data further includes at least one from the group consisting of: entity; entity and relationship; entity and attribute; and entity, relationship, and partnership set.

3. A computer system according to claim 2 further comprising at least one logic modeler, including PDL means for defining logical operations on at least one of said IFM design data and IM design data, said logical operations being defined by PDL design data.

4. A computer system according to claim 1 wherein said information modeler further comprises means for establishing an IM display having IM display objects representative of said IM design data, including means associated with said IM display selectively operative for externally applying signals to said IM display objects, whereby an externally applied signal associated with at least one of said IM display objects affects said IM design data; and

wherein said process modeler further comprises means for establishing a IFM display having IFM display objects representative of said IFM design data, including means associated with said IFM

- 5 display selectively operative for externally applying signals to said IFM display objects, whereby an externally applied signal associated with at least one of said IFM display objects affects said IFM design data.

10

5. A computer system according to claim 4 further comprising display linking means for dynamically correlating said IM display and said IFM display, whereby a change to any of said IM design
15 data affects a substantially immediate corresponding change in said IFM design data, and whereby a change to any of said IFM design data affects a substantially immediate change in said IFM design data.

20

6. A computer system according to claim 4 wherein said IFM display establishing means further comprises process explosion (PE) means for exploding at least one of said processes of said IFM design
25 data into zero, one or more process explosion (PE) processes;

- each of said PE processes including at least one of the group consisting of primitive processes and high-level (HL) processes, each of said HL
30 processes including one of the group consisting of primitive processes and HL processes; and

wherein each of said HL processes includes HL means for creating and modifying each of said PE processes.

35

7. A computer system according to claim 6 wherein said PE means further includes means for establishing a PE display of at least one of said primitive processes having PE display objects
5 representative of said IFM design data associated with said primitive process, and including means associated with said PE display selectively operative for externally applying signals to said PE display objects, whereby an externally applied signal
10 associated with at least one of said PE display objects affects said IFM design data; and
- wherein said PE display is selectively displayable simultaneous with at least one of said IM display, or said IFM display.

15

8. A computer system according to claim 4 wherein said design data includes at least two processes, and further comprising implosion means for consolidating at least two of said processes of said
20 design data into a process implosion (PI) process, each of said PI processes defining a high-level (HL) process.

9. A computer according to claim 8 wherein said
25 implosion means further includes means for establishing a PI display of at least one of said HL processes, and including means associated with said PI display selectively operative for externally applying signals to said PI display, whereby an
30 externally applied signal associated with said PI display affects said design data; and

wherein said PI display is selectively displayable simultaneous with at least one of said modeler displays.

5 10. A computer system according to claim 3 wherein said PDL design data is representative of at least one from the group consisting of text and tree structure diagram, and

10 wherein said operator means further include means for selectively displaying said text and tree structure diagrams.

11. A computer system according to claim 3
15 further comprising:

A. means for establishing an option modeler, having selectively operative OM means for creating, analyzing, and modifying syntactically
20 permitted options associated with said PDL design data, said OM means including:

(i) means for storing predetermined option criteria, each of said predetermined option criteria
25 being representative of at least one from the group consisting of semantic rules, logical relationships, expert rules, and heuristics;

(ii) means for comparing said PDL design
30 data with said predetermined option criteria; and

(iii) means for identifying correlated PDL design data, said correlated PDL design data being ones of said PDL design data which substantially match one of said predetermined criteria, said correlated PDL design data being representative of said syntactically permitted options;

B. means for establishing an options window having operator selectable option objects representative of said syntactically permitted options, whereby said syntactically permitted options are only ones of said PDL design data syntactically permitted at the time of establishment of said options window; and

15

C. means associated with said options window selectively operative for externally applying signals to said option modeler whereby an externally applied signal associated with at least one of said option objects affects said PDL design data.

12. A computer system according to claims 3 further comprising linking means for dynamically correlating said logic modeler, said information modeler and said process modeler, whereby a change to any of said PDL design data affects a corresponding PDL-IM/IFM change in said IM design data and said IFM design data, and whereby a change to any of said IM design data or said IFM design data affects a corresponding IM/IFM-PDL change in said PDL design data.

13. A computer system according to claims 3 and 12 wherein said operator means includes means for establishing at least one option facility and option objects therein, including:

5

means associated with said option facility selectively operative for externally applying signals to said information modeler, said process modeler, and said logic modeler whereby an externally applied
10 signal associated with at least one of said option objects affects said IM design data, said IFM design data and said PDL design data.

14. A computer system according to claim 3 or 12
15 wherein said operator means includes means for establishing at least one feedback window and feedback objects therein, including:

means associated with said feedback window
20 selectively operative to display data received from at least one of said information modeler, said process modeler, and said logic modeler.

15. A computer system according to claim 3 or 12
25 wherein said operator means includes means for establishing at least one diagram window and diagram objects therein, including:

means associated with said diagram window
30 selectively operative to display data received from at least one of said information modeler, said process modeler, and said logic modeler, defining said diagram objects;

means associated with said diagram window selectively operative for externally applying signals to said diagram objects, whereby an externally applied signal associated with at least one of said 5 diagram objects affects said IM design data, said IFM design data, and said PDL design data.

16. A computer system according to claim 3 or 12 wherein said externally applied signals have 10 associated therewith signal values, and wherein said information modeler, said information flow modeler, and said logic modeler comprises:

means for storing predetermined values, each 15 of said predetermined values being representative of at least one process, semantic rule, expert rule, logical relationship, heuristic, text field, or pin;

means for comparing said signal values with 20 said predetermined value;

means for identifying uncorrelated signals, said uncorrelated signals being ones of said signal values which do not substantially match one of said 25 predetermined values; and

means for generating flag signals representative of the identity of said uncorrelated signals.

-71-

17. A computer system according to claim 3 further comprising filter means for creating at least one from the group of subsets consisting of: an IM-subset of said information model; an IFM-subset of said process model; and, a PDL-subset of said PDL design data;

each of said IM-subset, IFM-subset, and PDL-subset being defined by operator-determined selection criteria, and being selected in accordance with said selection criteria from said IM design data, IFM design data, and PDL design data.

18. A computer system according to claim 17 wherein said filter means further comprises means for establishing an IM-subset display of said IM-subset having IM-subset display objects representative of said IM-subset, including means associated with said IM-subset display selectively operative for externally applying signals to said IM-subset display objects, whereby an externally applied signal associated with at least one of said IM-subset display objects affects said information model.

19. A computer system according to claim 17 wherein said filter means further comprises means for establishing an IFM-subset display of said IFM-subset having IFM-subset display objects representative of said IFM-subset, including means associated with said IFM-subset display selectively operative for externally applying signals to said IFM-subset display objects, whereby an externally applied signal associated with at least one of said IFM-subset display objects affects said process model.

20. A computer system according to claim 17 wherein said filter means further comprises means for establishing a PDL-subset display of said PDL-subset having PDL-subset display objects representative of
5 said PDL-subset, including means associated with said PDL-subset display selectively operative for externally applying signals to said PDL-subset display objects, whereby an externally applied signal associated with at least one of said PDL-subset
10 display objects affects said PDL design data.

21. A computer system according to claim 17 wherein said filter means further comprises:

15 means for establishing an IM-subset display of said IM-subset having IM-subset display objects representative of said IM-subset, including means associated with said IM-subset display selectively operative for externally applying signals to said
20 IM-subset display objects, whereby an externally applied signal associated with at least one of said IM-subset display objects affects said information model;

25 means for establishing an IFM-subset display of said IFM-subset having IFM-subset display objects representative of said IFM-subset, including means associated with said IFM-subset display selectively operative for externally applying signals to said
30 IFM-subset display objects, whereby an externally applied signal associated with at least one of said IFM-subset display objects affects said process model; and

means for establishing a PDL-subset display of said PDL-subset having PDL-subset display objects representative of said PDL-subset, including means associated with said PDL-subset display selectively operative for externally applying signals to said PDL-subset display objects, whereby an externally applied signal associated with at least one of said PDL-subset display objects affects said PDL design data.

10

22. A computer system according to claim 21 further comprising display linking means for dynamically correlating said IM-subset display, said IFM-subset display, and said PDL-subset display, wherein a change to any of said IM-subset design data affects a substantially immediate corresponding change in said IFM-subset design data and said PDL-subset design data, and wherein a change to any of said IFM-subset design data affects a substantially immediate corresponding change in said IM-subset design data and said PDL-subset design data, and wherein a change to any of said PDL-subset design data affects a substantially immediate corresponding change in said IM-subset design data and said IFM-subset design data.

23. A computer system according to claim 3 further comprising at least one function modeler, including FD design data representative of FD subprocesses, each of said subprocesses corresponding to at least one of said processes of said IFM design data, and having FD means for defining a hierarchy of said FD subprocesses of said IFM design data.

24. A computer system according to claim 23 further comprising filter means for creating at least one of the group of subsets consisting of: an IM-subset of said information model; an IFM-subset of
5 said process model; an FD-subset of said function model; and, a PDL-subset of said PDL design data;

each of said IM-subset, IFM-subset, FD-subset, and PDL-subset being defined by
10 operator-determined selection criteria, and being selected in accordance with said selection criteria from said IM design data, IFM design data, FD design data, and PDL design data.

15 25. A computer system according to claim 24 wherein said filter means further comprises:

means for establishing an IM-subset display of said IM-subset having IM-subset display objects
20 representative of said IM-subset, including means associated with said IM-subset display selectively operative for externally applying signals to said IM-subset display objects, whereby an externally applied signal associated with at least one of said
25 IM-subset display objects affects said information model;

means for establishing an IFM-subset display of said IFM-subset having IFM-subset display objects
30 representative of said IFM-subset, including means associated with said IFM-subset display selectively operative for externally applying signals to said IFM-subset display objects, whereby an externally

applied signal associated with at least one of said IFM-subset display objects affects said process model;

means for establishing an FD-subset display
5 of said FD-subset having FD-subset display objects representative of said FD-subset, including means associated with said FD-subset display selectively operative for externally applying signals to said FD-subset display objects, whereby an externally
10 applied signal associated with at least one of said FD-subset display objects affects said information model; and

means for establishing a PDL-subset display
15 of said PDL-subset having PDL-subset display objects representative of said PDL-subset, including means associated with said PDL-subset display selectively operative for externally applying signals to said PDL-subset display objects, whereby an externally
20 applied signal associated with at least one of said PDL-subset display objects affects said PDL design data.

26. A computer system according to claim 25
25 further comprising display linking means for dynamically correlating said IM-subset display, said IFM-subset display, said FD-subset display, and said PDL-subset display,

wherein a change to any of said IM-subset
30 design data affects a substantially immediate corresponding change in at least one from the group consisting of said IFM-subset design data, FD-subset design data, and said PDL-subset design data,

wherein a change to any of said IFM-subset design data affects a substantially immediate corresponding change in at least one from the group consisting of said IM-subset design data, said
5 FD-subset design data, and said PDL-subset design data,

wherein a change to any of said FD-subset design data affects a substantially immediate corresponding change in at least one from the group
10 consisting of said IM-subset design data, said IFM-subset design data, and said PDL-subset design data, and

wherein a change to any of said PDL-subset design data affects a substantially immediate
15 corresponding change in said IM-subset design data and said IFM-subset design data.

27. A computer system according to claim 23 wherein said hierarchy is representative of at least
20 one text or graphics, and

wherein said operator means further include means for selectively displaying said text and
25 graphics.

28. A computer system according to claim 27 further comprising linking means for dynamically correlating said function modeler, said logic modeler and said process modeler, whereby a change to any of
5 said FD design data affects a corresponding FD-PDL/IFM change in said PDL design data and said IFM design data, and whereby a change to any of said PDL design data or said IFM design data affects a corresponding PDL/IFM-FD change in said FD design
10 data.

29. A computer system according to claim 28 wherein said function modeler further comprises means for establishing a FD display having FD display
15 objects representative of said FD design data, including means associated with said FD display selectively operative for externally applying signals to said FD display objects, whereby an externally applied signal associated with at least one of said
20 FD display objects affects said FD design data; and

wherein said process modeler further comprises means for establishing a IFM display having IFM display objects representative of said IFM design
25 data, including means associated with said IFM display selectively operative for externally applying signals to said IFM display objects, whereby an externally applied signal associated with at least one of said IFM display objects affects said IFM
30 design data.

30. A computer system according to claim 29 further comprising display linking means for dynamically correlating said FD display and said IFM display, whereby a change to any of said FD design data affects a substantially immediate corresponding change in said IFM design data, and whereby a change to any of said IFM design data affects a substantially immediate change in said FD design data.

10 31. A computer system according to claim 30 wherein said IFM display establishing means further comprises process explosion (PE) means for exploding at least one of said processes of said IFM design data into zero, one or more process explosion
15 processes;

each of said PE processes including at least one of the group consisting of primitive processes and high-level (HL) processes, each of said HL processes including one of the group consisting of
20 primitive processes and HL processes; and

wherein each of said HL processes includes HL means for creating and modifying each of said PE processes.

32. A computer system according to claim 31 wherein said PE means includes means for establishing a PE display of at least one of said primitive processes having PE display objects representative of
5 said IFM design data associated with said primitive process, and including means associated with said PE display selectively operative for externally applying signals to said PE display objects, whereby an externally applied signal associated with at least
10 one of said PE display objects affects said IFM design data; and

wherein said PE display is selectively displayable simultaneous with at least one of said IM display, said IFM display, said PDL display, or said
15 FD display.

33. A computer system according to claim 3 further comprising:

20 A. means for establishing in said computer system one or more partnership sets within at least one IM design data, IFM design data, or PDL design data, each of said partnership sets being representative of zero, one or more partnerships;

25

B. means for establishing one or more of said partnerships, wherein each of said partnerships is characterized by one partnership set associated with itself or associated with one other partnership
30 set; and

C. means for accessing in said computer system one or more of said IM design data, said IFM design data, and said PDL design data by reference to a partnership set which is in partnership with
5 partnership sets of said one or more of said IM design data, said IFM design data, or said PDL design data.

34. A computer system according to claim 3
10 further comprising:

A. storage means for storing information representative of an ordered set of references to design data;

15

B. valuation means for assigning a data value to said ordered set, defining an entity-set variable; and

20 C. processing means selectively operable to process said entity-set variable in accordance with imposed criteria.

35. A computer system according to claim 3
25 wherein said logic modeler further comprises means for establishing a PDL display having PDL display objects representative of said PDL design data, including means associated with said PDL display selectively operative for externally applying signals
30 to said PDL display objects, whereby an externally applied signal associated with at least one of said PDL display objects affects said PDL design data.

36. A computer system according to claim 35 further comprising display linking means for dynamically correlating said IM display, said IFM display, and said PDL display, whereby a change to 5 any of said IM design data affects a substantially immediate corresponding change in said IFM design data and said PDL design data, and whereby a change to any of said IFM design data affects a substantially immediate change in said IFM design 10 data and said PDL design data, and whereby a change to any of said PDL design data affects a substantially immediate change in said IM design data and said IFM design data;

said IM design data being representative of 15 elements including at least one from the group consisting of attributes, partnership sets, keys, domains, dimensions, and allowable values.

37. A computer system according to claim 35 20 further comprising display linking means for dynamically correlating said IM display, said IFM display, and said PDL display, whereby a change to any of said IM design data affects a substantially immediate corresponding change in said IFM design 25 data and said PDL design data, and whereby a change to any of said IFM design data affects a substantially immediate change in said IFM design data and said PDL design data, and whereby a change to any of said PDL design data affects a 30 substantially immediate change in said IM design data and said IFM design data;

said IM design data being representative of elements including entities; and

said PDL design data being representative of elements including at least one from the group consisting of send and receive.

5 38. A computer system according to claim 1 further comprising at least one function modeler, including FD design data representative of FD subprocesses, each of said subprocesses corresponding to at least one of said processes of said IFM design
10 data, and having FD means for defining a hierarchy of said FD subprocess of said IFM design data.

39. An information management method for controlling information systems of organizations
15 comprising steps of:

A. establishing at least one information modeler including IM means for creating, analyzing and modifying an information model, said information
20 model having IM design data representative of elements including at least one from the group consisting of attribute, entity method, and partnership set, said attribute being independent of an entity;

25

B. establishing at least one process modeler including IFM means for creating, analyzing, and modifying a process model, said information flow model having IFM design data representative of
30 elements including at least one from the group consisting of processes, information storees, information flows, messages, sources, and destinations, said IFM design data corresponding to at least one of said elements of said IM design data;

35

C. establishing modeler linking means for dynamically correlating said information modeler and said process modeler, whereby a change to any of said IM design data affects a corresponding IM-IFM change 5 in said IFM design data, and whereby a change to any of said IFM design data affects a corresponding IFM-IM change in said IM design data; and

D. establishing operator means for 10 applying external signals to said information modeler and said process modeler, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said IM means and said IFM means.

15

40. A computer system for processing information representative of business transactions comprising:

A. two or more modelers including modeler 20 means for creating, analyzing, and modifying two or more models corresponding to each of said modelers, each of said models having design data;

B. at least one of said modelers being a 25 logic modeler, including PDL means for defining logical operations on at least one of said design data, said logical operations being defined by PDL design data;

C. operator means for applying external 30 signals to each of said modelers, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said modeler means; and

35

D. wherein said PDL design data corresponds to at least one from the group consisting of retrieve entity-set, send, receive, signal, when, reference to an entity method, reference to an entity-set, connect, and disconnect.

41. A computer system according to claim 40 wherein said PDL design data is representative of at least one from the group consisting of text and tree structure diagram, and

wherein said operator means further include means for selectively displaying said text and tree structure diagrams.

15

42. A computer system according to claim 41 wherein said text is representative of at least one from the group consisting of icons and alphanumeric symbols.

20

43. A computer system according to claim 42 wherein said operator means further includes means for externally applying signals to said logic modeler whereby an externally applied signal associated with at least one of said text affects said PDL design data.

44. A computer system according to claim 42 wherein said tree structure diagram includes nodes and pins, and wherein said nodes have text fields including data representative of at least one of logical operations, IM design data, or IFM design data.

30

45. A computer system according to claim 44 wherein said tree structure diagram includes nodes and pins, wherein said pins are representative of operation locations at said node, whereby an external
5 signal applied to at least one of said operation locations performs an operation at said operation location, said operation including at least one from the group consisting of: add data, replace data, delete data, add node, replace node, or delete node.
- 10 46. A computer system according to claim 40 further comprising at least one information modeler including IM means for creating, analyzing and modifying an information model, said information
15 model having IM design data representative of elements including at least one from the group consisting of attribute, entity method, attribute method, partnership set, and partnership set method, said attribute being independent of an entity.
- 20 47. A computer system according to claim 40 further comprising at least one process modeler including IFM means for creating, analyzing, and modifying a process model, said information flow
25 model having IFM design data representative of elements including at least one from the group consisting of processes, information stores, information flows, messages, views, process local stores, sources, and destinations, said IFM design
30 data relating to at least one of said elements of said IM design data.

48. A computer system according to claims 46 or 47 wherein said information modeler further comprises means for establishing an IM display having IM display objects representative of said IM design data, including means associated with said IM display selectively operative for externally applying signals to said IM display objects, whereby an externally applied signal associated with at least one of said IM display objects affects said IM design data;

10

wherein said process modeler further comprises means for establishing a IFM display having IFM display objects representative of said IFM design data, including means associated with said IFM display selectively operative for externally applying signals to said IFM display objects, whereby an externally applied signal associated with at least one of said IFM display objects affects said IFM design data; and

20

wherein said logic modeler further comprises means for establishing a PDL display having PDL display objects representative of said PDL design data, including means associated with said PDL display selectively operative for externally applying signals to said PDL display objects, whereby an externally applied signal associated with at least one of said PDL display objects affects said PDL design data.

30

49. A computer system according to claim 48 further comprising display linking means for dynamically correlating said IM display, said IFM display, and said PDL display whereby a change to any one of said IM design data, IFM design data, and PDL design data affects a substantially immediate corresponding change in at least one of said IM design data, IFM design data, and PDL design data.
- 10 50. A computer system according to claim 40 further comprising:
- A. storage means for storing information representative of an ordered set of references to
15 design data;
- B. valuation means for assigning a data value to said ordered set, defining an entity-set variable;
- 20 C. processing means selectively operable to process said entity-set variable in accordance with imposed criteria.
- 25 51. A computer system according to claim 50 wherein said design data includes entities.
52. A computer system according to claim 51 wherein said ordered set is ordered in accordance
30 with predetermined criteria.
53. A computer system according to claim 51 wherein said ordered set is ordered in accordance with operator selected criteria.
- 35

54. A computer system for establishing an adaptive window system in a dynamic model of information systems of organizations comprising:

5 A. one or more modelers including modeler means for creating, analyzing, and modifying one or more models corresponding to each of said modelers, each of said models having design data representative of elements including at least one from the group
10 consisting of attribute; attribute method; entity and attribute; entity method; entity and relationship; entity and entity method; entity, relationship and partnership set; process; view; process local store; information store; information flow; message; source;
15 and, destination;

B. at least one of said modelers including at least one logic modeler, having PDL means for defining logical operations on at least one of said
20 design data, said logical operations being defined by PDL design data, said PDL design data corresponding to at least one from the group consisting of insert, retrieve, update, delete, send, receive, signal, when, or reference to an entity method;

25 C. means for establishing an option modeler, having selectively operative OM means for creating, analyzing, and modifying syntactically permitted options associated with said PDL design
30 data, said OM means including:

(i) means for storing predetermined option criteria, each of said predetermined option criteria being representative of at least one from the group consisting of semantic rules, logical relationships, 5 expert rules, and heuristics;

(ii) means for comparing said PDL design data with said predetermined option criteria; and

10 (iii) means for identifying correlated PDL design data, said correlated PDL design data being ones of said PDL design data which substantially match one of said predetermined criteria, said correlated PDL design data being representative of 15 said syntactically permitted options;

D. means for establishing an options window having operator selectable option objects representative of said syntactically permitted 20 options, whereby said syntactically permitted options are only ones of said PDL design data syntactically permitted at the time of establishment of said options window; and

25 E. means associated with said options window selectively operative for externally applying signals to said option modeler whereby an externally applied signal associated with at least one of said option objects affects said PDL design data.

30

55. A computer system according to claim 54 further comprising at least one information modeler including IM means for creating, analyzing and modifying an information model, said information model having IM design data representative of elements including at least one from the group consisting of attribute, entity method, attribute method, partnership set, and partnership set method, said attribute being independent of an entity;

10

56. A computer system according to claim 55 further comprising at least one process modeler including IFM means for creating, analyzing, and modifying a process model, said information flow model having IFM design data representative of elements including at least one from the group consisting of processes, information stores, information flows, messages, views, process local stores, sources, and destinations, said IFM design data relating to at least one of said elements of said IM design data;

57. A computer system according to claim 54 wherein said operator means includes means for establishing at least one option facility and option objects therein, including:

means associated with said option facility selectively operative for externally applying signals to said information modeler, said process modeler, and said logic modeler whereby an externally applied signal associated with at least one of said option objects affects said IM design data, said IFM design data and said PDL design data.

35

58. A computer system according to claim 57 wherein said operator means includes means for establishing at least one feedback window and feedback objects therein, including:

5

means associated with said feedback window selectively operative to display data received from at least one of said information modeler, said process modeler, and said logic modeler.

10

59. A computer system according to claim 58 wherein said operator means includes means for establishing at least one diagram window and diagram objects therein, including:

15

means associated with said diagram window selectively operative to display data received from at least one of said information modeler, said process modeler, and said logic modeler, defining
20 said diagram objects;

means associated with said diagram window selectively operative for externally applying signals to said diagram objects, whereby an externally
25 applied signal associated with at least one of said diagram objects affects said IM design data, said IFM design data, and said PDL design data.

60. A computer system according to claim 59
30 wherein said externally applied signals have associated therewith signal values, and wherein said information modeler, said information flow modeler, and said logic modeler comprises:

means for storing predetermined values, each of said predetermined values being representative of at least one process, semantic rule, expert rule, logical relationship, heuristic, text field, or pin;

5

means for comparing said signal values with said predetermined value;

means for identifying uncorrelated signals,
10 said uncorrelated signals being ones of said signal values which do not substantially match one of said predetermined values; and

means for generating flag signals
15 representative of the identity of said uncorrelated signals.

61. A computer system according to claim 54 wherein said PDL design data corresponds to at least
20 one from the group consisting of insert, retrieve, update, delete, send, receive, signal, when, reference to an entity method, connect, or disconnect.

62. A computer system according to claim 61
25 wherein said PDL design data is representative of at least one from the group consisting of text and tree structure diagram, and

wherein said operator means further include
30 means for selectively displaying said text and tree structure diagrams.

63. A computer system according to claim 62 wherein said text is representative of at least one from the group consisting of icons and alphanumeric symbols.

5

64. A computer system according to claim 63 wherein said operator means further includes means for externally applying signals to said logic modeler whereby an externally applied signal associated with at least one of said text affects said PDL design data.

10

65. A computer system according to claim 62 wherein said tree structure diagram includes nodes and pins, and wherein said nodes have text fields including data representative of at least one of logical operations, IM design data, or IFM design data.

15

20 66. A computer system for specifying information representative of business transactions, comprising:

A. first storage means for storing transaction design data representative of the rules of a desired business transaction;

25

B. second storage means for storing user data in an original form;

30

5 C. processing means for reversibly
processing said user data whereby said
user data is transformed in accordance
with a predetermined processing
algorithm defined by said design data,
and associated means for making said
user data externally inaccessible
during said processing;

10 D. means internally operative on said
transformed user data for evaluating
said transformed user data in
accordance with predetermined logical
15 criteria defined by said design data;
and

upon identification of a first set of
conditions:

- 20 i. storing said transformed user
data;
- ii. making said stored
transformed user data
25 externally accessible; and

upon identification of a second set of
conditions:

- 30 i. inverse transforming said
user data to said original
form;

ii. storing said original form
user data; and

5 iii. making said stored original
form user data externally
accessible.

67. A computer system according to claim 66
wherein said user data is representative of elements
10 including at least one from the group consisting of:

 attribute; attribute method; entity method;
partnership set; partnership set method; entity;
entity and relationship; entity and attribute;
15 entity, relationship, and partnership set;

 process; information store; information
flow; message; view; process local store; source;
destination; event;

20

 keywords; identifiers; symbols; and,
statements.

68. A computer system according to claim 67
25 wherein said statements include at least one from the
group consisting of: insert; retrieve; update;
delete; send; receive; signal; when; connect;
disconnect; reference to an entity method; and,
reference to an entity.

30

69. A computer system according to claim 67
wherein said logical criteria includes at least one
from the group consisting of logical relationships,
expert rules, semantic rules, and heuristics.

35

70. A computer system according to claim 66 further comprising:

A. one or more modelers, and including
5 modeler means for creating, analyzing, and modifying
one or more models corresponding to each of said
modelers, each of said models having design data; and

B. operator means for applying external
10 signals to said modelers, said signals being
effective to evoke a creating, analyzing, or
modifying response from at least one of said modeler
means.

15 71. A computer system according to claim 70
wherein at least one of said modelers comprises a
logic modeler, including PDL means for defining
logical operations on at least one of said design
data, said logical operations being defined by PDL
20 design data.

72. A computer system according to claim 71
wherein said PDL design data corresponds to at least
one from the group consisting of insert, retrieve,
25 update, delete, send, receive, signal, when,
reference to an entity, connect, or disconnect.

73. A computer system according to claim 72 wherein said PDL design data is representative of at least one from the group consisting of text and tree structure diagram, and
- 5 wherein said operator means further include means for selectively displaying said text and tree structure diagrams.
- 10 74. A computer system according to claim 73 wherein said text is representative of at least one from the group consisting of icons and alphanumeric symbols.
- 15 75. A computer system according to claim 74 wherein said operator means further includes means for externally applying signals to said logic modeler whereby an externally applied signal associated with at least one of said text affects said PDL design
- 20 data.
76. A computer system according to claim 73 wherein said tree structure diagram includes nodes and pins, and wherein said nodes have text fields
- 25 including data representative of at least one of logical operations, IM design data, or IFM design data.

77. A computer system according to claim 76 wherein said tree structure diagram includes nodes and pins, wherein said pins are representative of operation locations at said node, whereby an external
5 signal applied to at least one of said operation locations performs an operation at said operation location, said operation including at least one from the group consisting of: add data, replace data, delete data, add node, replace node, or delete node.

10

78. A computer system according to claim 71 further comprising at least one information modeler including IM means for creating, analyzing and modifying an information model, said information
15 model having IM design data representative of elements including at least one from the group consisting of attribute, entity method, attribute method, partnership set, and partnership set method, said attribute being independent of an entity.

20

79. A computer system according to claim 78 further comprising at least one process modeler including IFM means for creating, analyzing, and modifying a process model, said information flow
25 model having IFM design data representative of elements including at least one from the group consisting of processes, information stores, information flows, messages, views, process local stores, sources, and destinations, said IFM design
30 data relating to at least one of said elements of said IM design data.

80. A computer system according to claim 79 wherein said information modeler further comprises means for establishing an IM display having IM display objects representative of said IM design data, including means associated with said IM display selectively operative for externally applying signals to said IM display objects, whereby an externally applied signal associated with at least one of said IM display objects affects said IM design data;
- 10 wherein said process modeler further comprises means for establishing a IFM display having IFM display objects representative of said IFM design data, including means associated with said IFM display selectively operative for externally applying signals to said IFM display objects, whereby an externally applied signal associated with at least one of said IFM display objects affects said IFM design data; and
- 20 wherein said logic modeler further comprises means for establishing a PDL display having PDL display objects representative of said PDL design data, including means associated with said PDL display selectively operative for externally applying signals to said PDL display objects, whereby an externally applied signal associated with at least one of said PDL display objects affects said PDL design data.
- 30

81. A computer system according to claim 80 further comprising display linking means for dynamically correlating said IM display, said IFM display, and said PDL display whereby a change to any one of said IM design data, IFM design data, and PDL design data affects a substantially immediate corresponding change in at least one of said IM design data, IFM design data, and PDL design data.
- 10 82. A computer system for processing complex business transactions comprising:
- A. at least one selectively operable outer subsystem including:
- 15
- i. first outer subsystem storage means for storing transaction design data representative of the rules of a desired business transaction;
- 20
- ii. second outer subsystem storage means for storing first user data in an original form;
- 25
- iii. processing means for reversibly processing said first user data whereby said first user data is transformed in accordance with a predetermined processing algorithm defined by said design data, and associated means for making said stored first user data externally inaccessible during said processing;
- 30

iv. means internally operative on said transformed first user data for evaluating said transformed first user data in accordance with predetermined logical criteria defined by said design data; and

upon identification of a first set of conditions:

(a) storing said transformed first user data;

(b) making said stored transformed first user data externally accessible; and

upon identification of a second set of conditions:

(a) inverse transforming said first user data to said original form;

(b) storing said original form first user data; and

(c) making said stored original form user data externally accessible;

B. at least one selectively operable nested inner subsystem including:

- 5 i. first inner subsystem storage means for storing nested transaction design data representative of the rules of a desired business transaction;
- 10 ii. second inner subsystem storage means for storing nested user data in an original form;
- 15 iii. processing means for reversibly processing said nested user data whereby said nested user data is transformed in accordance with a predetermined processing algorithm defined by said design data, and associated means for making said nested
20 user data externally inaccessible during said processing;
- 25 iv. means internally operative on said transformed nested user data for evaluating said transformed nested user data in accordance with predetermined logical criteria defined by said design data; and
30 upon identification of a first set of conditions:
 - 35 (a) storing said transformed nested user data;

- (b) making said stored transformed nested user data externally accessible; and

5 upon identification of a second set of conditions:

- 10 (a) inverse transforming said nested user data to said original form;
- (b) storing said original form user data; and
- 15 (c) making said stored original form nested user data accessible to said outer subsystem;

C. means for controlling in sequence:

- 20 i. said outer subsystem to be operative using said first user data;
- ii. said nested inner subsystem to be operative using said nested user data;
- 25 and
- iii. said outer subsystem to be operative using said accessible nested user data.

83. A computer system according to claim 82 wherein said first user data is representative of elements including at least one from the group consisting of:

5

attribute; attribute method; entity method; partnership set; partnership set method; entity; entity and relationship; entity and attribute; entity, relationship, and partnership set;

10

process; information store; information flow; message; view; process local store; source; destination; event;

15

keywords; identifiers; symbols; and, statements.

84. A computer system according to claim 82 wherein said nested user data is representative of 20 elements including at least one from the group consisting of:

attribute; attribute method; entity method; partnership set; partnership set method; entity; 25 entity and relationship; entity and attribute; entity, relationship, and partnership set;

process; information store; information flow; message; view; process local store; source; 30 destination; event;

keywords; identifiers; symbols; and, statements.

85. A computer system according to claim 82 further comprising:

A. storage means for storing information
5 representative of an ordered set of references to design data;

B. valuation means for assigning a data
value to said ordered set, defining an entity-set
10 variable;

C. processing means selectively operable
to process said entity-set variable in accordance
with imposed criteria.
15

86. A computer system according to claim 85 wherein said design data includes entities.

87. A computer system according to claim 86
20 wherein said ordered set is ordered in accordance with predetermined criteria.

88. A computer system according to claim 86 wherein said ordered set is ordered in accordance
25 with operator selected criteria.

89. A computer system according to claim 86 wherein said first user data includes entity-set
variables.
30

90. A computer system according to claim 86 wherein said nested user data includes entity-set
variables.

91. A computer system for establishing model subsets of a dynamic model of information systems of organizations comprising:

- 5 A. two or more modelers including modeler means for creating, analyzing, and modifying two or more models corresponding to each of said modelers, each of said models having design data representative of elements including at least one from the group
10 consisting of:

 attribute; entity and attribute; attribute method; entity and relationship; entity method; entity and entity method; entity, relationship and
15 partnership set; process; view; process local store; information store; information flow; message; source; and, destination;

- B. operator means for applying external
20 signals to each of said modelers, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said modeler means;

- 25 C. modeler linking means for dynamically correlating said design data, whereby a change to any of said design data of one of said modelers affects a substantially immediate corresponding change to said design data of others of said modelers; and
30

D. filter means for creating at least one model subset of at least one of said models, each of said model subsets being defined by operator-determined selection criteria, and being
5 selected in accordance with said selection criteria from each of said design data.

92. A computer system according to claim 91 wherein said filter means further comprises means for
10 establishing at least one model subset display of each of said model subsets, and having model subset display objects representative of at least one of said model subsets, including means associated with each of said model subset displays selectively
15 operative for externally applying signals to each of said model subset display objects, whereby an externally applied signal associated with at least one of said model subset display objects affects at least one associated one of said models.

20 93. A computer system according to claim 92 further comprising display linking means for dynamically correlating each of said model subset displays, wherein a change to any of said model
25 subset design data affects a substantially immediate corresponding change in others of said model subset design data.

94. A computer system for establishing model subsets of a dynamic model of information systems of organizations comprising:

- 5 A. at least one information modeler including IM means for creating, analyzing and modifying an information model, said information model having IM design data;
- 10 B. at least one process modeler including IFM means for creating, analyzing, and modifying a process model, said information flow model having IFM design data representative of elements including at least one from the group consisting of processes,
15 information stores, information flows, messages, views, process local stores, sources, and destinations, said IFM design data relating to at least one of said elements of said IM design data;
- 20 C. at least one logic modeler, including PDL means for defining logical operations on at least one of said IFM design data and IM design data, said logical operations being defined by PDL design data;

D. modeler linking means for dynamically correlating said information modeler, said process modeler, and said logic modeler, wherein a change to any of said IM design data affects a corresponding
5 IM-IFM/PDL change in said IFM design data and said PDL design data, and wherein a change to any of said IFM design data affects a corresponding IFM-IM/PDL change in said IM design data and said PDL design data, and wherein a change to any of said PDL design
10 data affects a corresponding PDL-IM/IFM change in said IM design data and said IFM design data;

E. operator means for applying external signals to said information modeler and said process
15 modeler, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said IM means and said IFM means; and

F. filter means for creating at least one
20 of the group of subsets consisting of an IM-subset of said information model, an IFM-subset of said process model, and a PDL-subset of said PDL design data, each of said IM-subset, IFM-subset, and PDL-subset being defined by operator-determined selection criteria,
25 and being selected in accordance with said selection criteria from said IM design data, IFM design data, and PDL design data.

95. A computer system according to claim 94 wherein said IM design data is representative of elements including at least one from the group consisting of: attribute; attribute method; partnership set; partnership set method; entity; entity method; entity and relationship; entity and attribute, and; entity, relationship, and partnership set.
- 10 96. A computer system according to claim 95 wherein said filter means further comprises:
- means for establishing an IM-subset display of said IM-subset having IM-subset display objects
- 15 representative of said IM-subset, including means associated with said IM-subset display selectively operative for externally applying signals to said IM-subset display objects, whereby an externally applied signal associated with at least one of said
- 20 IM-subset display objects affects said information model;
- means for establishing an IFM-subset display of said IFM-subset having IFM-subset display objects
- 25 representative of said IFM-subset, including means associated with said IFM-subset display selectively operative for externally applying signals to said IFM-subset display objects, whereby an externally applied signal associated with at least one of said
- 30 IFM-subset display objects affects said process model; and

means for establishing a PDL-subset display of said PDL-subset having PDL-subset display objects representative of said PDL-subset, including means associated with said PDL-subset display selectively
5 operative for externally applying signals to said PDL-subset display objects, whereby an externally applied signal associated with at least one of said PDL-subset display objects affects said PDL design data.

10

97. A computer system according to claim 96 further comprising display linking means for dynamically correlating said IM-subset display, said IFM-subset display, and said PDL-subset display,
15 wherein a change to any of said IM-subset design data affects a substantially immediate corresponding change in said IFM-subset design data and said PDL-subset design data, and wherein a change to any of said IFM-subset design data affects a
20 substantially immediate corresponding change in said IM-subset design data and said PDL-subset design data, and wherein a change to any of said PDL-subset design data affects a substantially immediate corresponding change in said IM-subset design data
25 and said IFM-subset design data.

98. A computer system according to claim 94 wherein said PDL design data corresponds to at least one from the group consisting of insert, retrieve,
30 update, delete, send, receive, signal, when, reference to an entity method, connect, and disconnect.

99. A computer system according to claim 94 further comprising at least one function modeler, including FD design data representative of FD subprocesses, each of said subprocesses corresponding to at least one of said processes of said IFM design data, and having FD means for defining a hierarchy of said FD subprocesses of said IFM design data.

100. A computer system according to claim 99 further comprising linking means for dynamically correlating said function modeler, said logic modeler and said process modeler, whereby a change to any of said FD design data affects a corresponding FD-PDL/IFM change in said PDL design data and said IFM design data, and whereby a change to any of said PDL design data or said IFM design data affects a corresponding PDL/IFM-FD change in said FD design data.

101. A computer system according to claim 100 wherein said function modeler further comprises means for establishing a FD display having FD display objects representative of said FD design data, including means associated with said FD display selectively operative for externally applying signals to said FD display objects, whereby an externally applied signal associated with at least one of said FD display objects affects said FD design data; and

wherein said process modeler further comprises means for establishing a IFM display having IFM display objects representative of said IFM design data, including means associated with said IFM display selectively operative for externally applying signals to said IFM display objects, whereby an externally applied signal associated with at least one of said IFM display objects affects said IFM design data.

10

102. A computer system according to claim 101 further comprising display linking means for dynamically correlating said FD display and said IFM display, whereby a change to any of said FD design data affects a substantially immediate corresponding change in said IFM design data, and whereby a change to any of said IFM design data affects a substantially immediate change in said FD design data.

20 103. A computer system for establishing subprocesses in a model of information management systems of organizations, comprising:

A. at least one modeler including modeler means for creating, analyzing, and modifying one or more models corresponding to each of said modelers, each of said models having design data, said design data including at least one process;

30 B. operator means for applying external signals to each of said modelers, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said modeler means;

35

C. means for establishing at least one modeler display having display objects representative of said design data, including means associated with said display selectively operative for externally
5 applying signals to said display objects, whereby an externally applied signal associated with at least one of said display objects affects associated ones of said design data;

10 D. process explosion (PE) means for exploding at least one of said processes of said design data into zero, one or more process explosion (PE) processes;

each of said PE processes including at least
15 one of the group consisting of primitive processes and high-level (HL) processes, each of said HL processes including one of the group consisting of primitive processes and HL processes; and

wherein each of said HL processes includes
20 HL means for creating and modifying each of said PE processes; and

E. means for establishing a PE display of at least one of said primitive processes having PE
25 display objects representative of said design data associated with said primitive process, and including means associated with said PE display selectively operative for externally applying signals to said PE display objects, whereby an externally applied signal
30 associated with at least one of said PE display objects affects said design data; and

wherein said PE display is selectively displayable simultaneously with at least one of said modeler displays.

104. A computer system according to claim 103 wherein said design data includes at least two processes, and further comprising implosion means for consolidating at least two of said processes of said design data into a process implosion (PI) process, each of said PI processes defining a high-level (HL) process.

105. A computer system according to claim 104 wherein said implosion means further includes means for establishing a PI display of at least one of said HL processes, and including means associated with said PI display selectively operative for externally applying signals to said PI display, whereby an externally applied signal associated with said PI display affects said design data; and

wherein said PI display is selectively displayable simultaneous with at least one of said modeler displays.

20

106. Display apparatus comprising:

A. means for establishing at least one set of one or more objects, wherein at least one of said objects is representative of a subset of one or more sub-objects;

B. means for selectively displaying said objects of said set in a structured set array;

30

C. means for selectively displaying said sub-objects of said subset in a structured subset array, wherein said structured subset array is selectively displayable simultaneous with said structured set array;

D. means associated with said structured set array selectively operative for externally applying signals to said objects, whereby an externally applied signal associated with at least one of said objects affects said sub-objects; and

E. means associated with said structured subset array selectively operative for externally applying signals to said sub-objects, whereby an externally applied signal associated with at least one of said sub-objects affects said objects of said set.

20 107. A computer system for establishing subprocesses in a model of information management systems of organizations, comprising:

A. at least one modeler including modeler means for creating, analyzing, and modifying one or more models corresponding to each of said modelers, each of said models having design data, said design data including at least two processes;

30 B. operator means for applying external signals to each of said modelers, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said modeler means;

35

- C. means for establishing at least one modeler display having display objects representative of said design data, including means associated with said display selectively operative for externally
5 applying signals to said display objects, whereby an externally applied signal associated with at least one of said display objects affects associated ones of said design data; and
- 10 D. implosion means for consolidating at least two of said processes of said design data into a process implosion (PI) process, each of said PI processes defining a high-level (HL) process.
- 15 108. A computer system according to claim 107 wherein said implosion means further includes means for establishing a PI display of at least one of said HL processes, and including means associated with said PI display selectively operative for externally
20 applying signals to said PI display, whereby an externally applied signal associated with said PI display affects said design data; and
wherein said PI display is selectively displayable simultaneous with at least one of said
25 modeler displays.
109. A computer system for information management systems of organizations, comprising:
- 30 A. one or more modelers including modeler means for creating, analyzing, and modifying one or more models corresponding to each of said modelers, each of said models having design data;

B. means for establishing in said computer system one or more partnership sets within at least one of said design data, each of said partnership sets being representative of zero, one or more
5 partnerships;

C. means for establishing one or more of said partnerships, wherein each of said partnerships is characterized by one partnership set associated
10 with itself or associated with one other partnership set;

D. means for accessing in said computer system one or more of said design data by reference
15 to a partnership set which is in partnership with partnership sets of said one or more of said design data;

E. operator means for applying external
20 signals to each of said modelers, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said modeler means; and

F. modeler linking means for dynamically correlating said design data, whereby a change to any
25 of said design data of one of said modelers affects a substantially immediate corresponding change to said design data of others of said modelers.
30

110. A computer system according to claim 109 wherein said design data is representative of elements including at least one from the group consisting of:

5

attribute; attribute method; entity and attribute; entity and relationship; entity and entity method; entity, relationship and partnership set; process; view; process local store; information
10 store; information flow; message; event; source; and, destination.

111. A computer system according to claim 110 wherein at least one modeler comprises an information
15 modeler, including IM means for creating, analyzing and modifying an information model,

said information model having IM design data representative of elements including at least one from the group consisting of: attribute; entity
20 method; attribute method; partnership set; and, partnership set method, said attribute being independent of an entity.

112. A computer system according to claim 111 wherein at least one modeler comprises a process modeler, including IFM means for creating, analyzing, and modifying a process model,

5 said information flow model having IFM design data representative of elements including at least one from the group consisting of: processes; information stores; information flows; messages; views; process local stores; events; sources; and, 10 destinations,

said IFM design data relating to at least one of said elements of said IM design data;

113. A computer system according to claim 112 15 further comprising modeler linking means for dynamically correlating said information modeler and said process modeler, wherein a change to any of said IM design data affects a corresponding IM-IFM change in said IFM design data, and wherein a change to any 20 of said IFM design data affects a corresponding IFM-IM change in said IM design data.

114. A computer system according to claim 113 wherein said information modeler further comprises 25 means for establishing an IM display having IM display objects representative of said IM design data, including means associated with said IM display selectively operative for externally applying signals to said IM display objects, whereby an externally 30 applied signal associated with at least one of said IM display objects affects said IM design data; and

wherein said process modeler further comprises means for establishing a IFM display having IFM display objects representative of said IFM design data, including means associated with said IFM display selectively operative for externally applying signals to said IFM display objects, whereby an externally applied signal associated with at least one of said IFM display objects affects said IFM design data.

10

115. A computer system according to claim 114 further comprising display linking means for dynamically correlating said IM display and said IFM display, whereby a change to any of said IM design data affects a substantially immediate corresponding change in said IFM design data, and whereby a change to any of said IFM design data affects a substantially immediate change in said IFM design data.

20

116. A computer system for an information management system, comprising:

A. storage means for storing information representative of an ordered set of references to user data;

B. valuation means for assigning a data value to said ordered set, defining an entity-set variable;

30

C. processing means selectively operable to process said entity-set variable in accordance with imposed criteria.

35

117. A computer system according to claim 116 wherein said design data includes entity instances.

118. A computer system according to claim 117
5 wherein said ordered set is ordered in accordance with predetermined criteria.

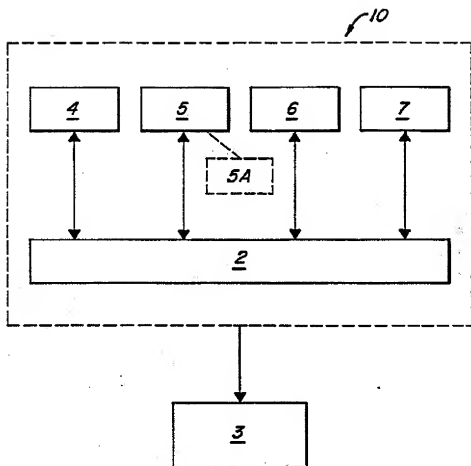
119. A computer system according to claim 117 wherein said ordered set is ordered in accordance
10 with operator selected criteria.

120. A computer system according to claim 117 further comprising:

15 A. at least one modeler including modeler means for creating, analyzing, and modifying one or more models corresponding to each of said modelers, each of said models having design data, said design data including at least one process; and

20 B. operator means for applying external signals to each of said modelers, said signals being effective to evoke a creating, analyzing, or modifying response from at least one of said modeler
25 means.

1/37

**FIG. 1**

2/37

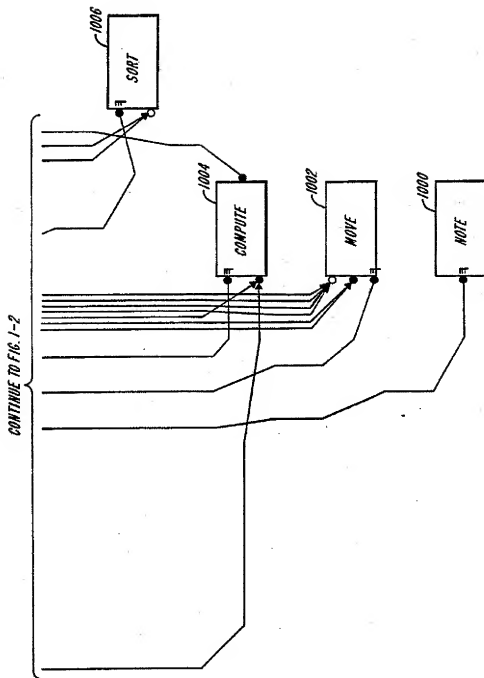
FIG. 1-23		FIG. 1-24	
FIG. 1-19	FIG. 1-20	FIG. 1-21	FIG. 1-22
FIG. 1-14	FIG. 1-15	FIG. 1-16	FIG. 1-18
FIG. 1-9	FIG. 1-10	FIG. 1-11	FIG. 1-13
FIG. 1-6	FIG. 1-7	FIG. 1-8	
FIG. 1-3	FIG. 1-4	FIG. 1-5	
		FIG. 1-2	
		FIG. 1-1	

FIG. 1A

3/37

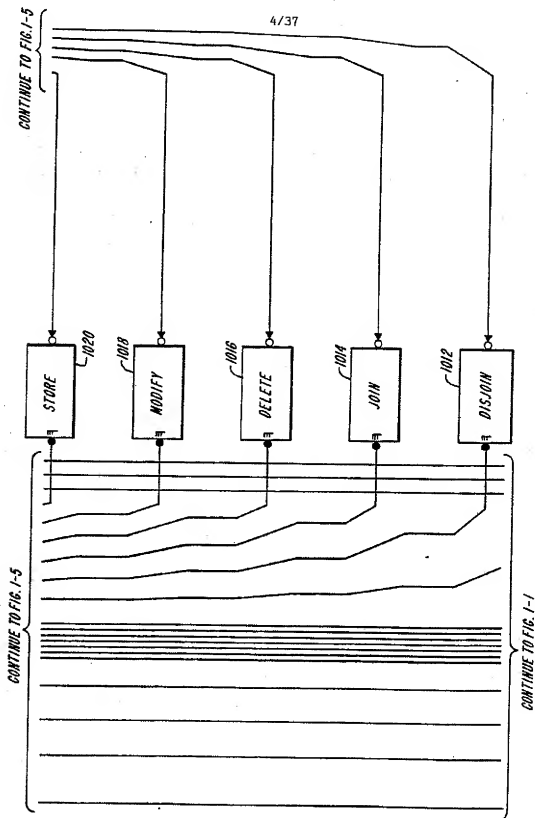
1989-11-21
8001

1010
GWBACHMAN

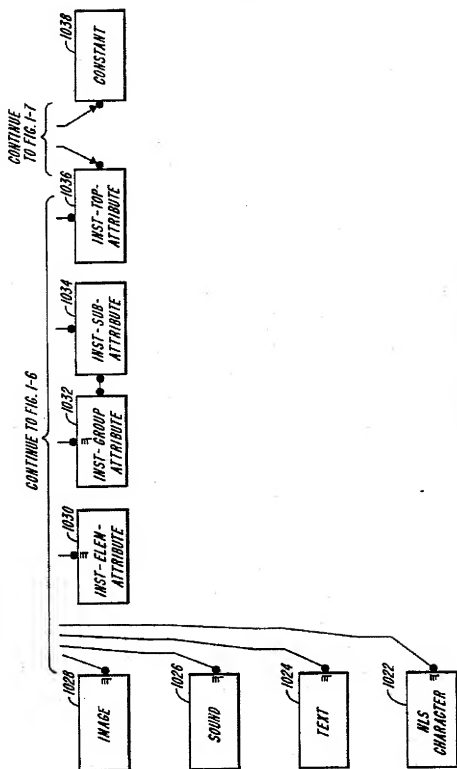


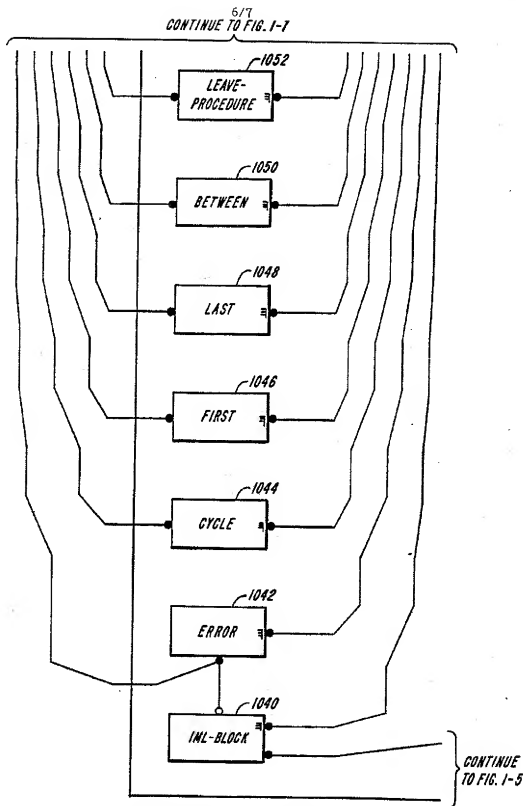
CONTINUE TO FIG. 1-2

SUBSTITUTE SHEET

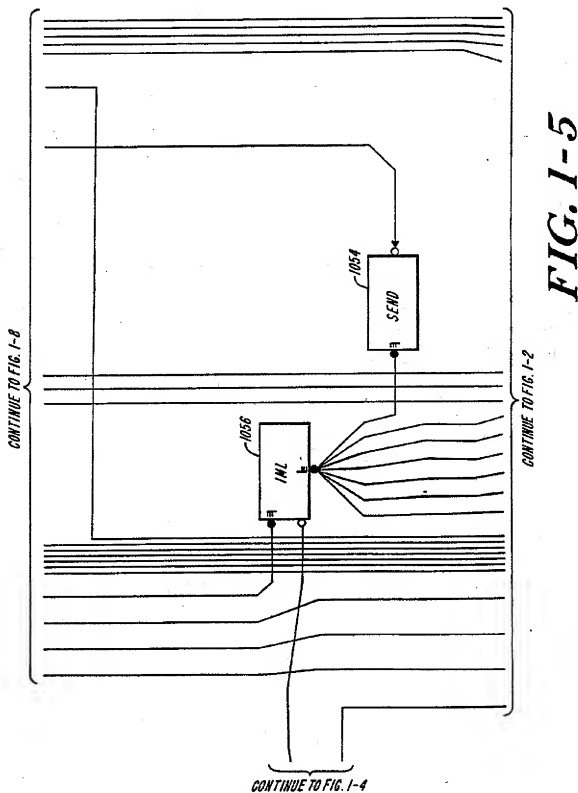
**FIG. 1-2**

5/37

**FIG. 1-3**

**FIG. 1-4**

7/37



SUBSTITUTE SHEET

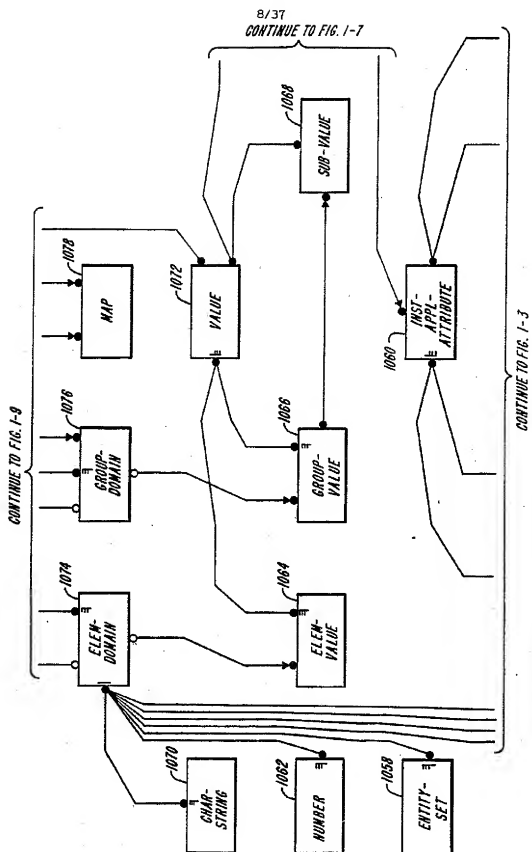
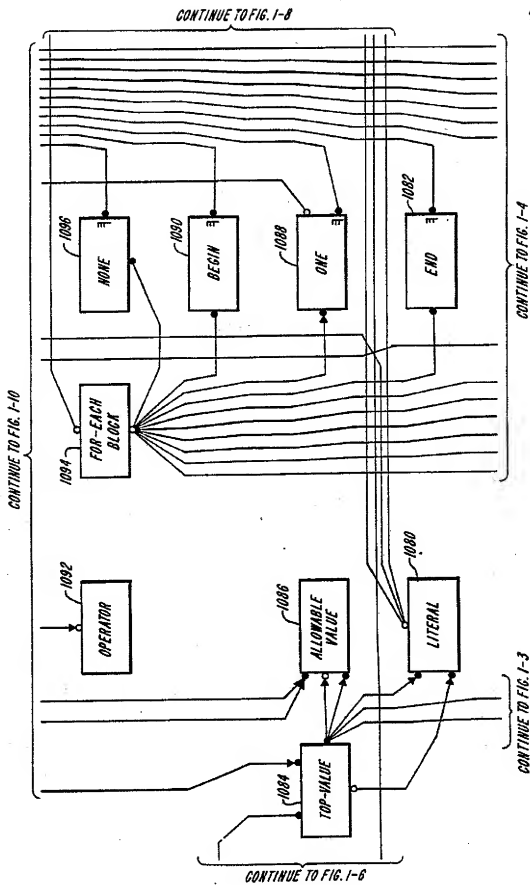
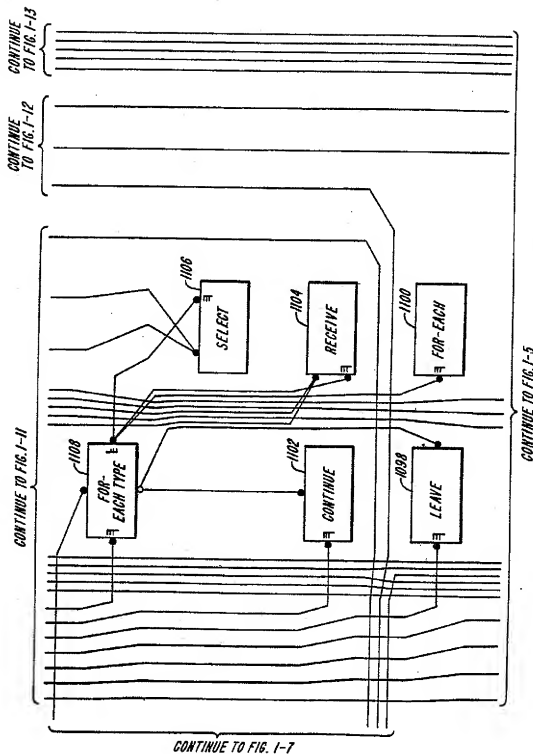


FIG. 1-6

9/37

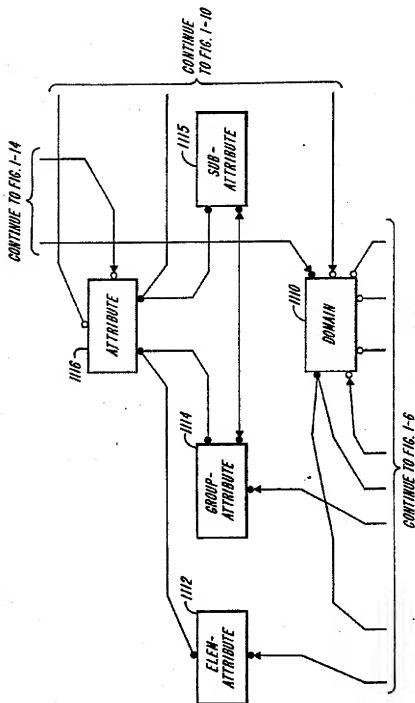
**FIG. 1-7**

10/37



SUBSTITUTE SHEET

11/37

**FIG. 1-9**

12/37

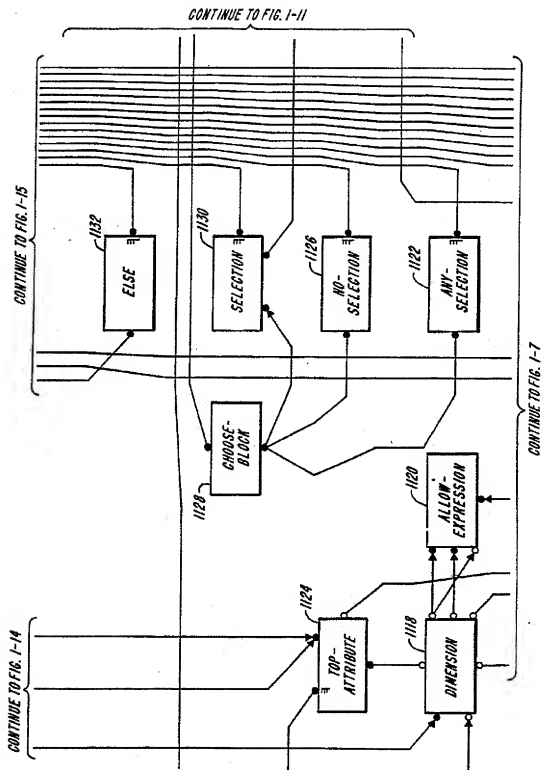


FIG. 1-10

13/37

CONTINUE TO FIG. 1-12

CONTINUE TO FIG. 1-16

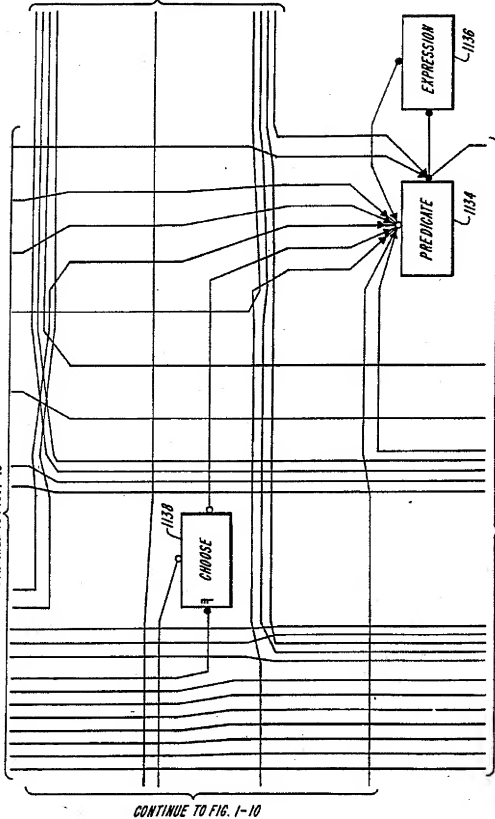


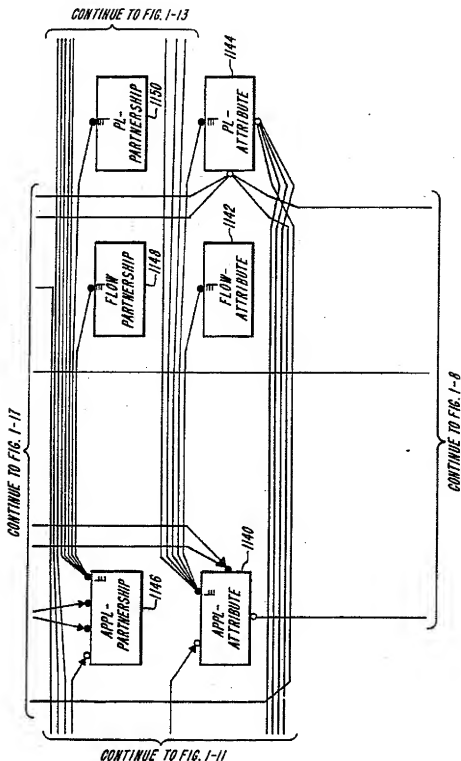
FIG. 1-11

CONTINUE TO FIG. 1-8

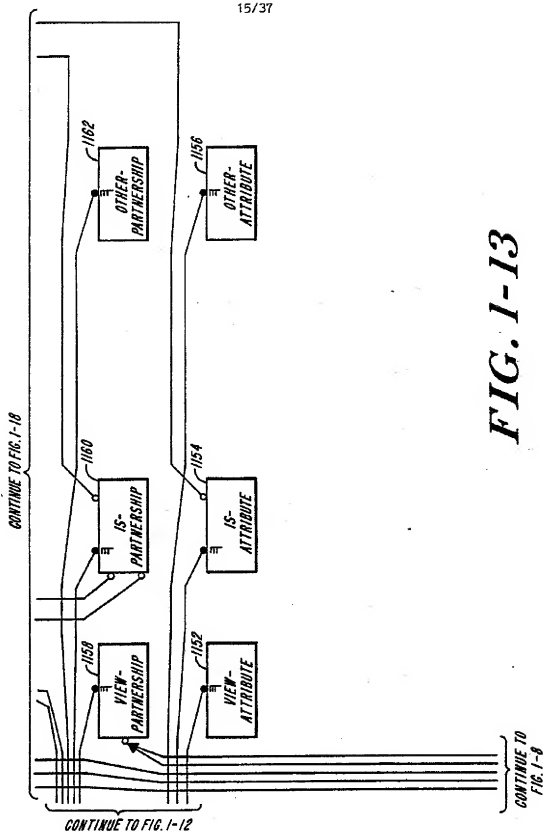
CONTINUE TO FIG. 1-10

SUBSTITUTE SHEET

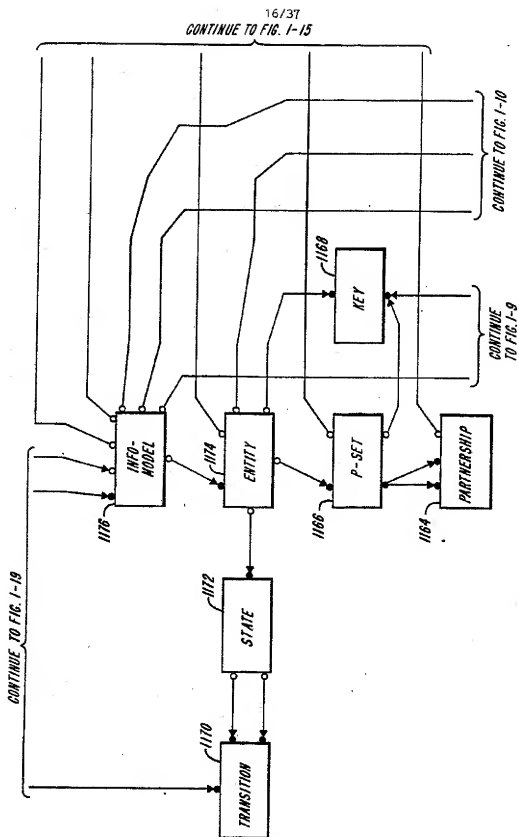
14/37

**FIG. 1-12**

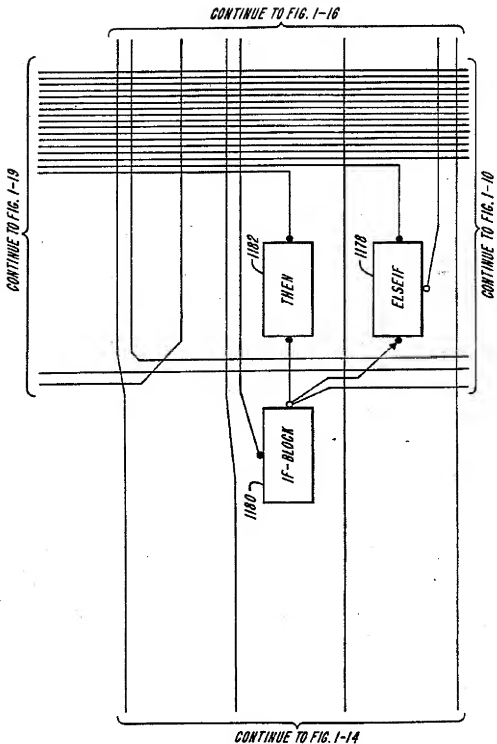
15/37



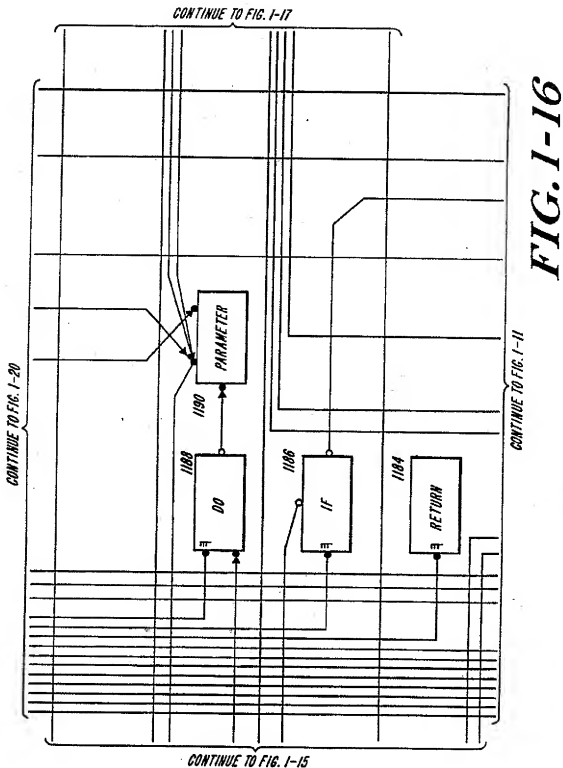
SUBSTITUTE SHEET

**FIG. 1-14**

17737

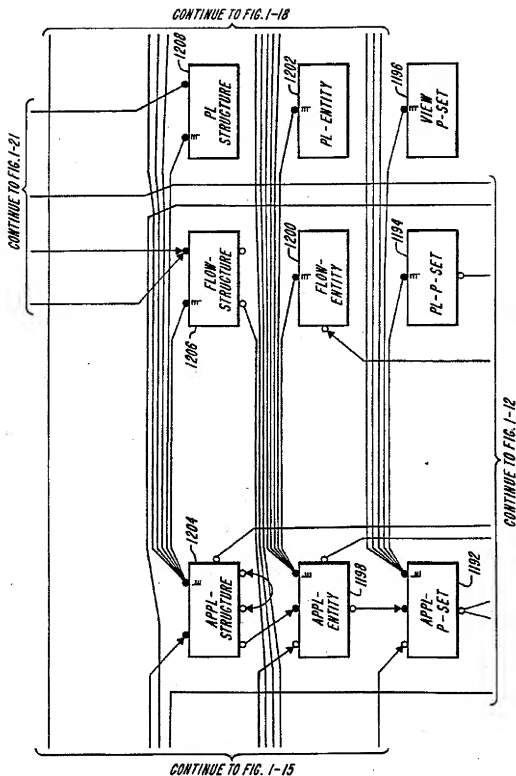
FIG. 1-15**SUBSTITUTE SHEET**

18/37

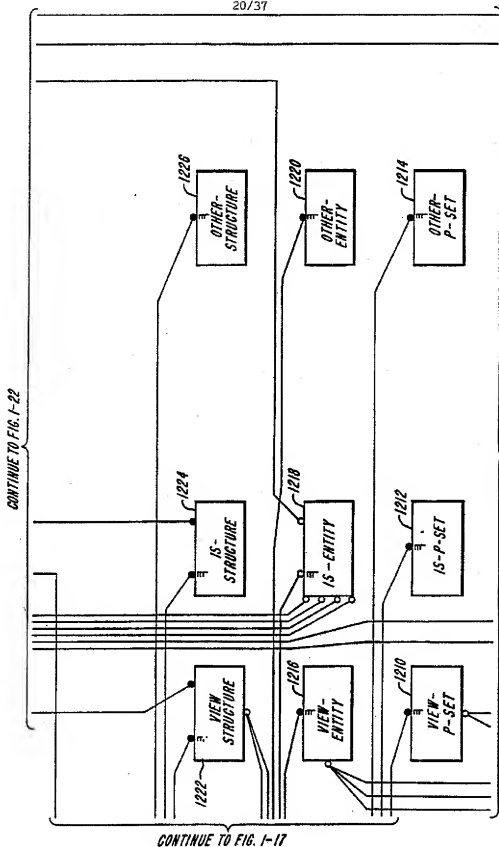


SUBSTITUTE SHEET

19/37

**FIG. 1-17****SUBSTITUTE SHEET**

20/37

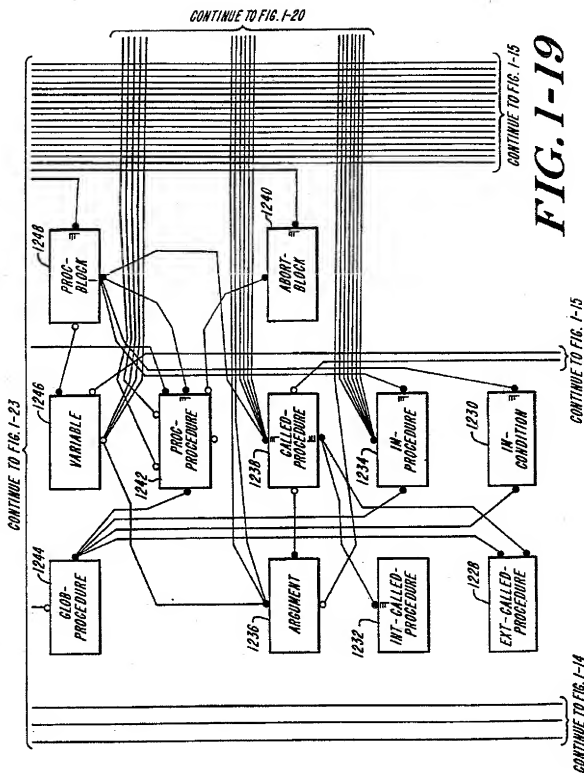


CONTINUE TO FIG. 1-22

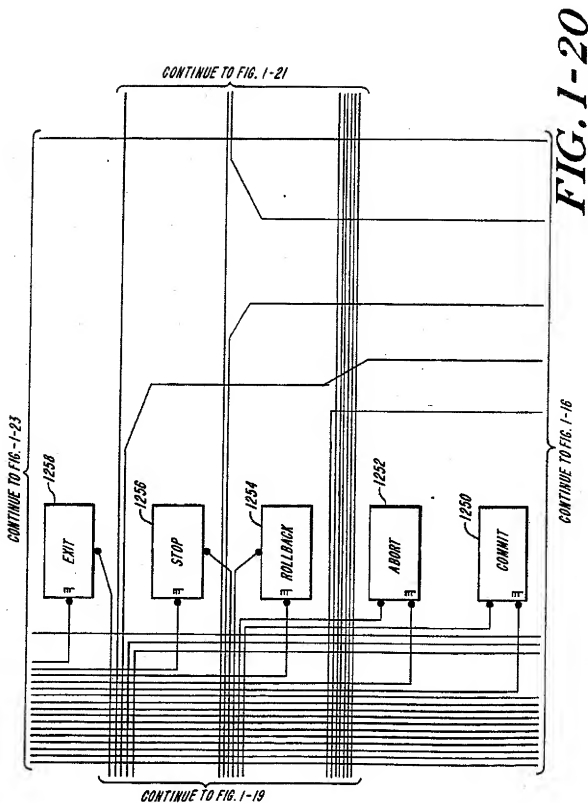
CONTINUE TO FIG. 1-13

CONTINUE TO FIG. 1-17

21/37

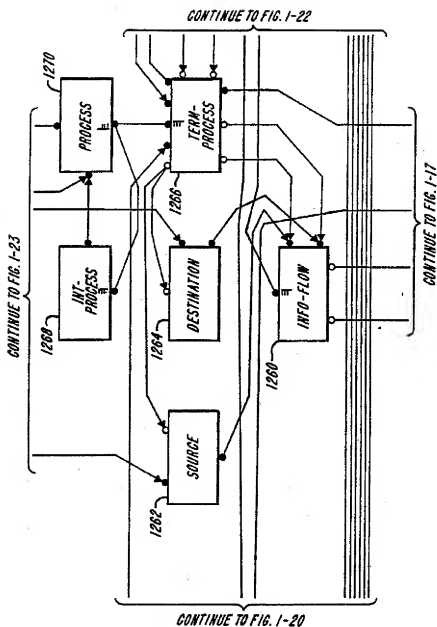


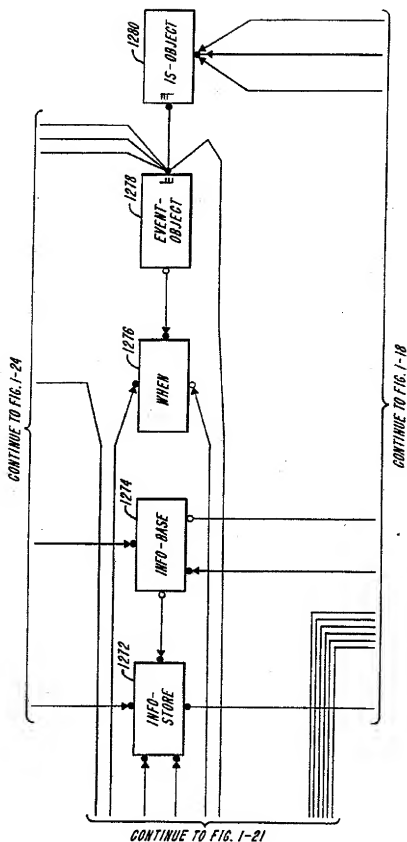
22/31

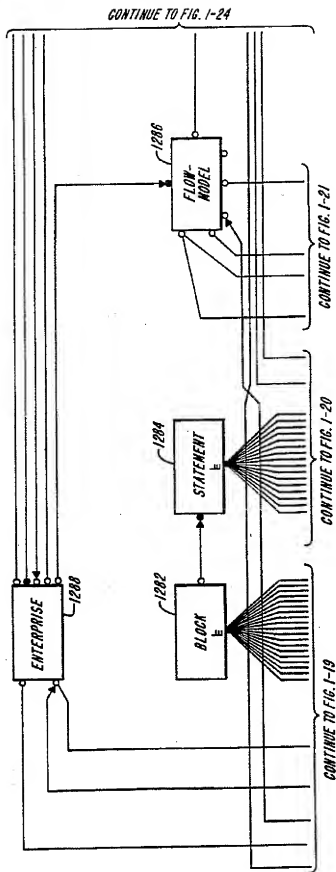


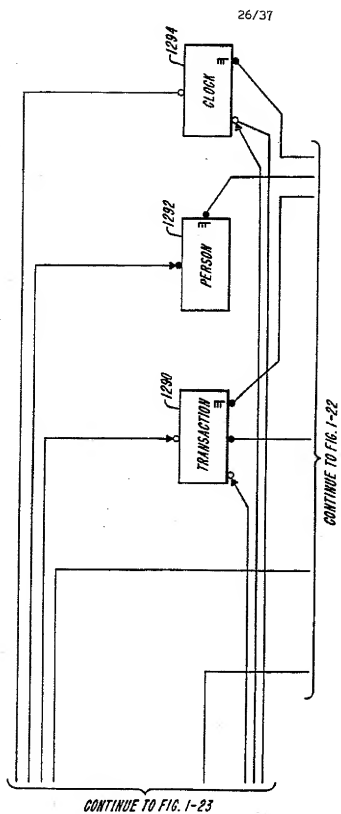
SUBSTITUTE SHEET

23/37

**FIG. 1-21**

**FIG. 1-22**

**FIG. 1-23**

**FIG. 1-24**

27/37

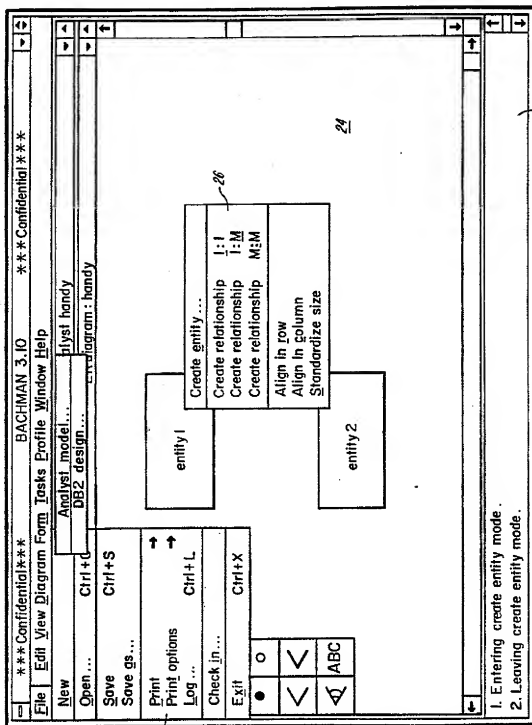


FIG. 2

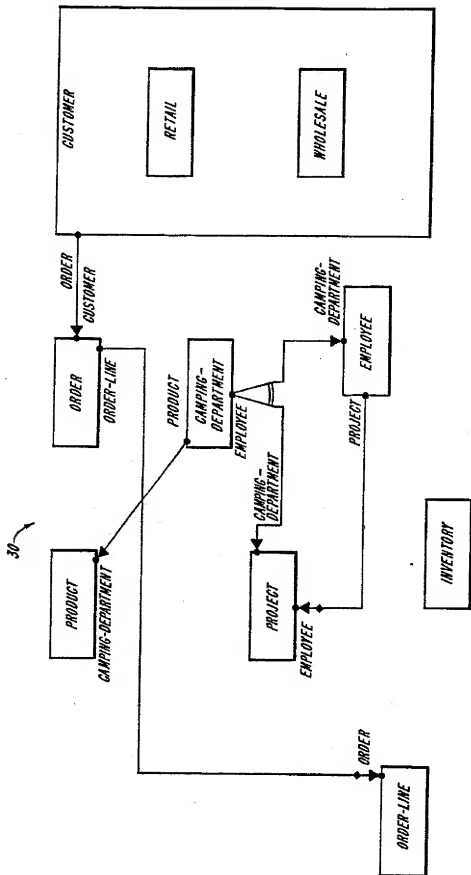


FIG. 3

29/37

*** Confidential *** BACHMAN 3.10 *** Confidential ***

File Edit View Diagram Form Tasks Profile Window Help

Analyst handy

E - R diagram: handy

Create entity in handy information model

Mask:

Description:

Expected volume:

Entities (name, expected volume)

Apply OK Add attr. Cancel Help

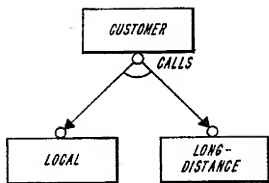
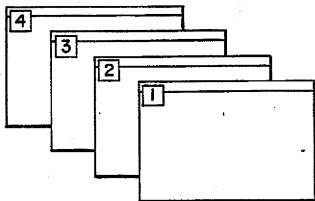
24

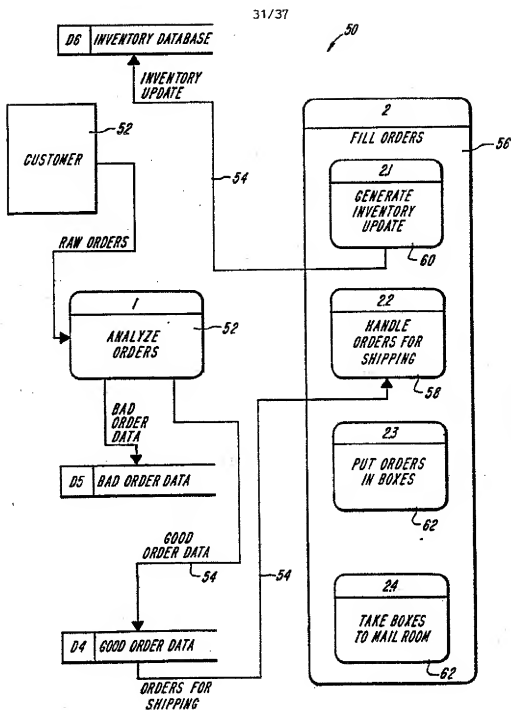
40

1. Entering create entity mode.
2. Leaving create entity mode.

FIG. 3A

30/37

**FIG. 4A****FIG. 4B**

**FIG. 5**

32/37

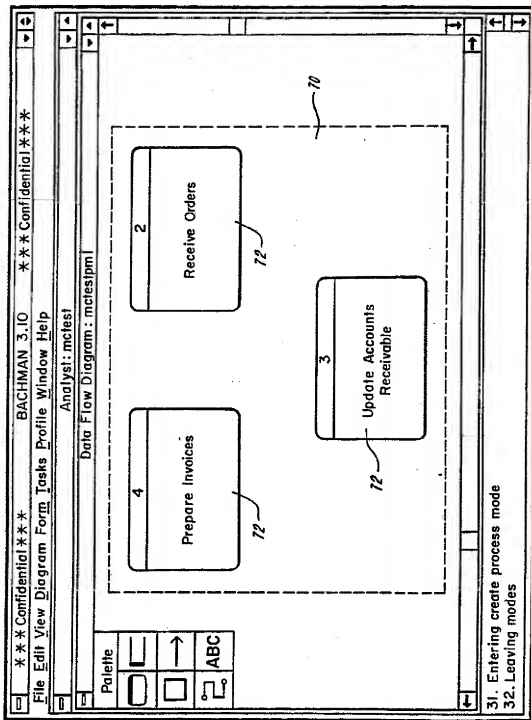


FIG. 6A

33/37

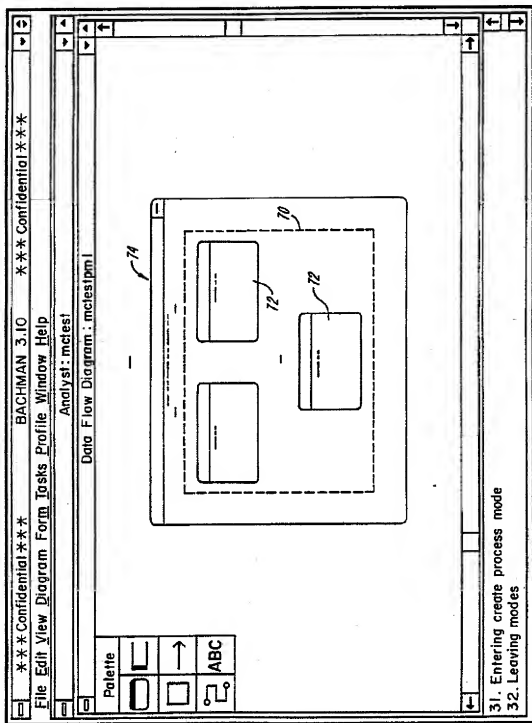
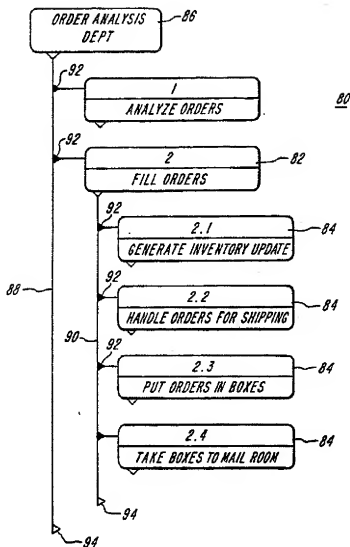


FIG. 6B

34/37

**FIG. 7**

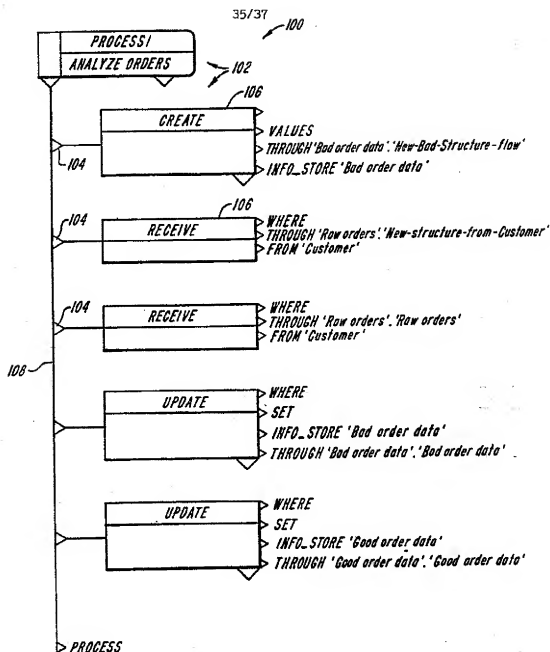
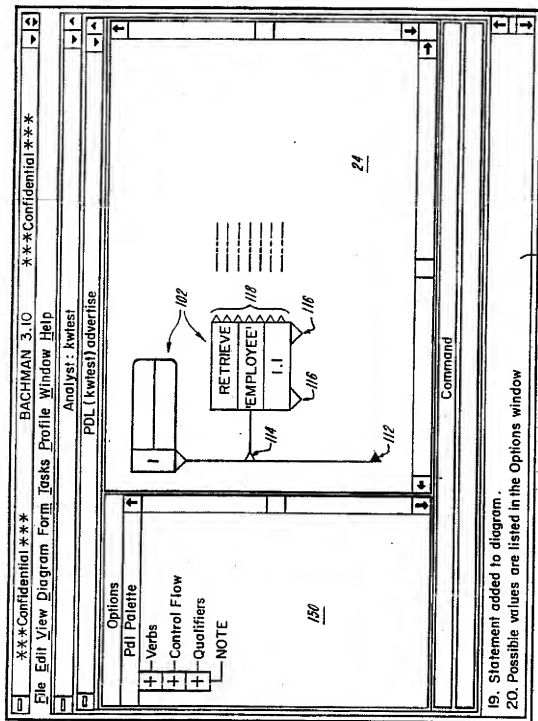


FIG. 8

36/37



19. Statement added to diagram.
 20. Possible values are listed in the Options window

FIG. 9

37/37

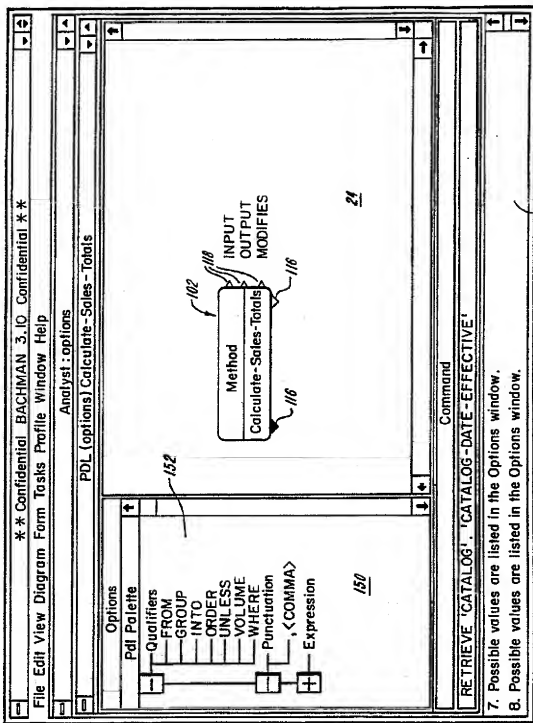


FIG. 10

INTERNATIONAL SEARCH REPORT

International Application No. **PCT/US91/02890**

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all) *

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC(5): **G06F/ 3/03; 9/00; 15/18**

U.S. CL: **364/200,513,521: 340/700**

II. FIELDS SEARCHED

Minimum Documentation Searched *

Classification System	Classification Symbols
U.S.	364/200, 364/900, 364/521, 340/700, 364/513 "MS FILE"

Documentation Searched other than Minimum Documentation
to the Extent that such Documents are Included in the Fields Searched *

COMMERCIAL DATABASES SEARCHED i.e. DIALOG, INSPEC.

III. DOCUMENTS CONSIDERED TO BE RELEVANT *

Category *	Citation of Document, ** with indication, where appropriate, of the relevant passages †‡	Relevant to Claim No. †§
Y	COMPUTER-AIDED SOFTWARE ENGINEERING, FIRST EDITION MARCH 1988, CHRIS GANE, RAPID SYSTEM DEVELOPMENT INC. (See pages 1-1 to 7-9)	1-16, 23-32, 35-39, 46-60, 66-82
Y	COMPUTER-AIDED SOFTWARE ENGINEERING, 1990, CHRIS GANE PRENTICE HALL INC. (See pages 3-30, 33-38 and 61-86)	1-16, 23-32, 35-39, 46-60, 66-82
Y	PC TECH JOURNAL, VOL.: V6 ISSUE: n9, SEPTEMBER 1988, ANDREW TOPPER, "BUILDING UP TO IEW/WS" pages 110-123.	1-109, 116-120
Y	PC TECH JOURNAL, VOL.: V6 ISSUE: n8, AUGUST 1988 CARMA MCCLURE, "THE CASE FOR STRUCTURAL DEVELOPEMENT" pages 52-67.	1-39

* Special categories of cited documents: †

"A" document defining the general state of the art which is not
considered to be of particular relevance

"E" earlier document but published on or after the international
filing date

"L" document which may throw doubt on priority claim(s) or
which is cited to establish the publication date of another
citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or
other means

"P" document published prior to the international filing date but
later than the priority date claimed

"T" later document published after the international filing date
or priority date and not in conflict with the application but
cited to understand the principle or theory underlying the
invention

"X" document of particular relevance; the claimed invention
cannot be considered novel or cannot be considered to
involve an inventive step

"Y" document of particular relevance; the claimed invention
cannot be considered to involve an inventive step when the
document is combined with one or more other such docu-
ments, such combination being obvious to a person skilled
in the art.

"A" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
12 JULY 1991	20 AUG 1991
International Searching Authority	Signature of Authorized Officer
ISA/US	<i>MEHMET GECKTI</i>

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category*	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No
A	Communications of the ACM Vol.: 32 Issue: n8, August 1989, Tobby J. Teorey, Guangping Wei, Deborah L. Bolton and John A. Koenig, "ER Model Clustering as an Aid for User Communication and Documentation in Database Design". pages 975-987.	1-120
Y	PC TECH JOURNAL, Vol.: V6 Issue: n8, August 1988 Andrew Topper, "Excelling with Case" pages 71-79	1-16, 23-32 35-39, 40-60, 66-82
Y	US, A, 4,631,664, (BACHMAN) 23 DECEMBER 1986 (See entire document)	109-115
Y	US, A, 4,841,441 (NIXON) 20 JUNE 1989 (See column 13, 17-18 and 9-10, 5-6 and 7-8)	1-9, 66-69
A	US, A, 4,894,771 (KUNITI) 16 JANUARY 1990 (See Fig. 2-4 and col. 3-5)	1-120
Y	US, A, 4,774,661 (KUMPATI) 27 SEPTEMBER 1988 (See Figs. 4-10) and Cols. 7-12	1-9

FURTHER INFORMATION CONTINUED FROM THE SECOND SHEET

V. ☐ OBSERVATIONS WHERE CERTAIN CLAIMS WERE FOUND UNSEARCHABLE *

This international search report has not been established in respect of certain claims under Article 17(2) (a) for the following reasons:

1. ☐ Claim numbers because they relate to subject matter ¹² not required to be searched by this Authority, namely:

2. ☐ Claim numbers because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out ¹³, specifically:

3. ☐ Claim numbers because they are dependent claims not drafted in accordance with the second and third sentences of PCT Rule 6.4(a).

VI. ☒ OBSERVATIONS WHERE UNITY OF INVENTION IS LACKING *

This International Searching Authority found multiple inventions in this international application as follows:

- | | |
|-------------------|----------------------|
| I. Claims 1-30 | V. Claims 91-102 |
| II. Claims 40-53 | VI. Claims 103-108 |
| III. Claims 54-65 | VII. Claims 109-115 |
| IV. Claims 66-90 | VIII. Claims 116-120 |

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims of the international application. See attached telephone memorandum
2. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims of the international application for which fees were paid, specifically claims:

3. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim numbers:

4. ☐ As all searchable claims could be searched without effort justifying an additional fee, the international Searching Authority did not invite payment of any additional fee.

Remark on Protest

- ☐ The additional search fees were accompanied by applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.

TELEPHONE MEMORANDUM ATTACHMENT

- I. Claims 1-39 drawn to a dynamic model of information systems of organizations classified in class 364 subclass (200);
- II. Claims 40-53 drawn to a computer system for processing information representative of business transactions classified in class 364 subclass (900);
- III. Claims 54-65 drawn to an adaptive window system classified in class 364 subclass (521);
- IV. Claims 66-90 drawn to a computer system for specifying information representative of business transactions classified in class 364 subclass (401);
- V. Claims 91-102 drawn to a computer system for establishing model subsets of a dynamic model of information systems of organizations classified in class 364 subclass (513); (continued of next page)
- VI. Claims 103-108 drawn to a computer system for establishing sub-processes in a model of information management systems of organizations classified in class 364 subclass (513);
- VII. Claims 109-115 drawn to a computer system for information management systems of organization classified in class 364 subclass (200); and
- VIII. Claims 116-120 drawn to a computer system for information management systems classified in class 364 subclass (900).

